Pronunciation Clustering and Modeling of Variability for Appearance-Based Sign Language Recognition

Morteza Zahedi, Daniel Keysers, and Hermann Ney

Lehrstuhl für Informatik VI, Computer Science Department,
RWTH Aachen University, D-52056 Aachen, Germany
{zahedi, keysers, ney}@informatik.rwth-aachen.de

Abstract. In this paper, we present a system for automatic sign language recognition of segmented words in American Sign Language (ASL). The system uses appearance-based features extracted directly from the frames captured by standard cameras without any special data acquisition tools. This means that we do not rely on complex preprocessing of the video signal or on an intermediate segmentation step that may produce errors. We introduce a database for ASL word recognition extracted from a publicly available set of video streams. One important property of this database is the large variability of the utterances for each word. To cope with this variability, we propose to model distinct pronunciations of each word using different clustering approaches. Automatic clustering of pronunciations improves the error rate of the system from 28.4% to 23.2%. To model global image transformations, the tangent distance is used within the Gaussian emission densities of the hidden Markov model classifier instead of the Euclidean distance. This approach can further reduce the error rate to 21.5%.

1 Introduction

In the domain of sign language recognition from video, most approaches try to segment and track the hands and head of the signer in a first step and subsequently extract a feature vector from these regions [1, 2, 3, 4]. Segmentation can be difficult because of possible occlusions between the hands and the head of the signer, noise or brisk movements. Many approaches therefore use special data acquisition tools like data gloves, colored gloves or wearable cameras. These special tools may be difficult to use in practical situations.

In this work, we introduce a database of video streams for American sign language (ASL) word recognition. The utterances are extracted from a publicly available database and can therefore be used by other research groups. This database, which we call 'BOSTON50', consists of 483 utterances of 50 words. One important property of this database is the large visual variability of utterances for each word. This database is therefore more difficult to recognize automatically than databases in which all utterances are signed uniformly. So far, this problem has not been dealt with sufficiently in the literature on sign language recognition.

To overcome these shortcomings we suggest the following novel approaches:

S. Gibet, N. Courty, and J.-F. Kamp (Eds.): GW 2005, LNAI 3881, pp. 68–79, 2006.

1. The system presented in this paper is designed to recognize sign language words using simple appearance-based features extracted directly from the frames which are captured by standard cameras without any special data acquisition tools. This means that we do not rely on complex preprocessing of the video signal or on an intermediate segmentation step that may produce errors.
2. Because of the high variability of utterances of the same class, we explicitly model different pronunciations of each word of the database. We employ and compare different clustering methods to determine the partitioning into pronunciations: manual clustering, k-means clustering, and hierarchical LBG-clustering. Manual clustering uses a hand-labeled partitioning of the utterances. The k-means algorithm is initialized with the number of clusters and manually selected seed utterances. The hierarchical LBG-clustering partitions the data automatically and only needs one parameter to control the coarseness of the clustering. The results obtained lead us to also consider a nearest neighbor classifier that performs surprisingly well.
3. To deal with the image variability, we model global affine transformations of the images using the tangent distance [6] within the Gaussian emission densities instead of the Euclidean distance.

In Sections 2 and 3, we introduce the database BOSTON50 and the appearance-based features used in the system, respectively. Section 4 describes the decision making and the hidden Markov model (HMM) classifier. Tangent distance and the way it is employed in the HMM is explained in Section 5. In Section 6, the different clustering methods and their properties are described. Finally, the experimental results and conclusions are discussed in Sections 7 and 8.

2 Database

The National Center for Sign Language and Gesture Resources of the Boston University has published a database of ASL sentences[1] [7]. It consists of 201 annotated video streams of ASL sentences. Although this database was not recorded primarily for image processing and recognition research, we considered it as a starting point for a recognition corpus because the data are available to other research groups and, thus, can be a basis for comparisons of different approaches.

The signing is captured simultaneously by four standard stationary cameras where three of them are black/white and the remaining one is a color camera. Two black/white cameras, directed towards the signer's face, form a stereo pair. Another camera is installed on the side of the signer. The color camera is placed between the cameras of the stereo pair and is zoomed to capture only the face of the signer. The movies are recorded at 30 frames per second and the size of the frames is 312×242 pixels. We use the published video streams at the same frame rate but extract the upper center part of size 195×165 pixels. (Parts of the bottom of the frames show some information about the frame and the left and right border of the frames are unused.)

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

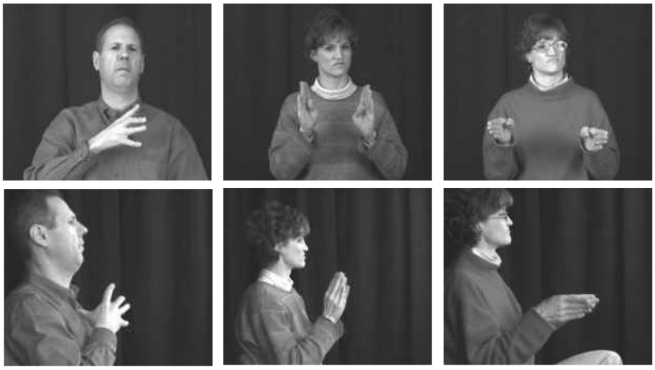

Fig. 1. The signers as viewed from the two camera perspectives

To create our database for ASL word recognition which we call BOSTON50, we extracted 483 utterances of 50 words from this database as listed in the appendix along with the number of utterances of each word. The utterances of the sign language words are segmented within our group manually.

In the BOSTON50 database, there are three signers, one of them male and two female. The signers are dressed differently and the brightness of their clothes is different. We use the frames captured by two of the four cameras, one camera of the stereo camera pair in front of the signer and the lateral camera. Using both of the stereo cameras and the color camera may be useful in stereo and facial expression recognition, respectively. Both of the cameras used are in fixed positions and capture the videos simultaneously. The signers and the views of the cameras are shown in Figure 1.

3 Feature Extraction

In this section, we briefly introduce the appearance-based features used in our ASL word recognition system. In [5], we introduce different appearance-based features in more detail, including the original image, skin color intensity, and different kinds of first- and second-order derivatives. The results show that down-scaled original images extracted after skin intensity thresholding perform very well. According to these results we employ these features in the work presented here.

The definition of the features is based on basic methods of image processing. The features are directly extracted from the images of the video frames. We denote by $Y_t(i, j)$ the pixel intensity at position (i, j) in the frame t of a sequence, $t = 1, \ldots, T$.

To disregard background pixels, we use a simple intensity thresholding. This thresholding aims at extracting the hand and the head, which form brighter regions in the images. This approach is not a perfect segmentation and we cannot rely on it easily for tracking the hands because the output of the thresholding consists of the two hands, face and possibly some parts of the signer's clothes.

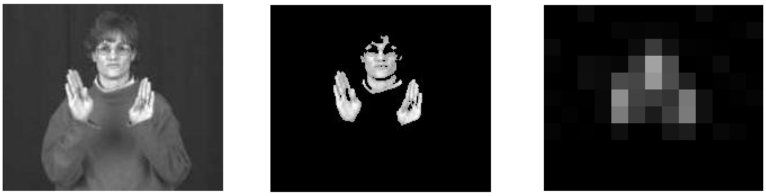

Fig. 2. Example of the features used by the classifier: original image (left), thresholded image (center), and down-scaled image (right)

$$X_t(i,j) = \begin{cases} Y_t(i,j) & : \quad Y_t(i,j) > \Theta \\ 0 & : \quad \text{otherwise} \end{cases} \tag{1}$$

Where $X_t(i,j)$ is an image frame at time t with the brightness threshold Θ.

We can transfer the matrix of an image to a vector x_t and use it as a feature vector. To decrease the size of the feature vector, we use the original image down-scaled to 13×11 pixels denoted by X'_t.

$$x_{t,d} = X'_t(i,j), \quad d = 13 \cdot j + i, \tag{2}$$

where $x_t = [x_{t,1}, ..., x_{t,d}, ..., x_{t,D}]$ is the feature vector at time t with the dimension $D = 143$.

Some examples of features after processing are shown in Figure 2. To increase the information extracted from the videos, we may use the frames of two cameras. One of the cameras is installed in front of the signer and the second one is fixed at one side. We concatenate the information of the frames captured simultaneously by these cameras. We weight the features extracted by the two cameras because there is more occlusion of the hands in the images captured by the lateral camera. According to experiments reported in [5], we weight the features of the front camera and lateral camera with the weights 0.38 and 0.62, respectively.

4 Decision Process

The decision making of our system employs HMMs to recognize the sign language words[2]. This approach is inspired by the success of the application of HMMs in speech recognition [8] and also most sign language recognition systems [1, 2, 3, 4, 5]. The recognition of sign language words is similar to spoken word recognition in the modelling of sequential samples. The topology of the HMM used is shown in Figure 3. There is a transition loop at each state and the maximum allowed transition is set to two, which means that, at most, one state can be skipped. We consider one HMM for each word $w = 1, ..., W$. The basic decision rule used for the classification of $x_1^T = x_1, ..., x_t, ... x_T$ is:

[2] Some of the code used in feature extraction and decision making is based on the LTI library that is available under the terms of the GNU Lesser General Public License at http://ltilib.sourceforge.net.

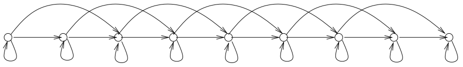

Fig. 3. The topology of the employed HMM

$$x_1^T \longrightarrow r(x_1^T) = \arg\max_w \left(Pr(w|x_1^T)\right)$$
$$= \arg\max_w \left(Pr(w) \cdot Pr(x_1^T|w)\right) , \tag{3}$$

where $Pr(w)$ is the prior probability of class w, and $Pr(x_1^T|w)$ is the class conditional probability of x_1^T given class w. The $Pr(x_1^T|w)$ is defined as:

$$Pr(x_1^T|w) = \max_{s_1^T} \prod_{t=1}^{T} Pr(s_t|s_{t-1}, w) \cdot Pr(x_t|s_t, w), \tag{4}$$

where s_1^T is the sequence of states, and $Pr(s_t|s_{t-1}, w)$ and $Pr(x_t|s_t, w)$ are the transition probability and emission probability, respectively. The transition probability is estimated by simple counting. We use the Gaussian mixture densities as emission probability distribution $Pr(x_t|s_t, w)$ in the states. The emission probability is defined as:

$$Pr(x_t|s_t, w) = \sum_{l=1}^{L(s_t,w)} Pr(x_t, l|s_t, w)$$
$$= \sum_{l=1}^{L(s_t,w)} Pr(l|s_t, w) \cdot Pr(x_t|s_t, w, l), \tag{5}$$

where $L(s_t, w)$ is the number of densities in each state and

$$Pr(x_t|s_t, w, l) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma^2_{l,s_t,w,d}}} \cdot \exp\left(-\frac{(x_{t,d} - \mu_{l,s_t,w,d})^2}{\sigma^2_{l,s_t,w,d}}\right). \tag{6}$$

In this work, the sum is approximated by the maximum, and the emission probability is defined as:

$$Pr(x_t|s_t, w) = \max_l Pr(x_t, l|s_t, w)$$
$$= \max_l Pr(l|s_t, w) \cdot Pr(x_t|s_t, w, l). \tag{7}$$

To estimate $Pr(x_t|s_t, w)$, we use the maximum likelihood estimation method for the parameters of the Gaussian distribution, i.e. the mean $\mu_{s_t,w,d}$ and the variances $\sigma_{s_t,w,d}$. Here, the covariance matrix is modeled to be diagonal, i.e. all off-diagonal elements are fixed at zero. The number of states for the HMM of each word is determined by the minimum sequence length of the training samples. Instead of a density-dependent estimation of the variances, we use pooling during the training of the HMM, which means that we do not estimate variances for each density of the HMM, but instead we estimate one set of variances for all densities in the complete model (word-dependent pooling).

We use the Viterbi algorithm to find the maximizing state sequence s_1^T. In the Viterbi algorithm, we calculate the score of the observation feature vector x_t in the emission probability distribution $Pr(x_t|s_t,w)$ at each state s_t. Assuming the Gaussian function with diagonal covariances for $Pr(x_t|s_t,w)$, as described above, this score is calculated as:

$$-\log Pr(x_t|s_t,w) = \min_l \Big\{ \underbrace{\frac{1}{2}\sum_{d=1}^{D} \frac{(x_{t,d}-\mu_{l,s_t,w,d})^2}{\sigma^2_{l,s_t,w,d}}}_{\text{distance}} - \log Pr(l|s_t,w) + \frac{1}{2}\sum_{d=1}^{D} \log(2\pi\sigma^2_{l,s_t,w,d}) \Big\}. \tag{8}$$

In this work, the feature vector x_t is a down-scaled image at time t with a dimensionality of $D = 143$. Therefore, the sum $\sum_{d=1}^{D}(x_{t,d}-\mu_{l,s_t,w,d})^2/\sigma^2_{l,s_t,w,d}$ is the distance between the observation image at time t and the mean image $\mu_{l,s_t,w}$ of the state s_t which is scaled by the variances $\sigma^2_{l,s_t,w,d}$. This scaled Euclidean distance can be replaced by other distance functions such as the tangent distance, which we will introduce in the following section.

The number of utterances in the database for each word is not large enough to separate them into training and test sets, for example some words of the database occur only twice. Therefore, we employ the leaving one out method for training and classification, i.e. we test the classifier on each sample in turn while training on the remaining 482 samples. The percentage of the misclassified utterances is the error rate of the system.

5 Tangent Distance

In this section, we give an overview of the distance measure invariant to affine transformations called *tangent distance*, which was first introduced in [9]. The incorporation into a statistical system was presented in [6]. An invariant distance measure ideally takes into account transformations of the patterns, yielding small values for patterns which mostly differ by a transformation that does not change class-membership.

Let $x_t \in \mathbb{R}^D$ be a pattern, and $x_t(\alpha)$ denote a transformation of x_t that depends on a parameter L-tuple $\alpha \in \mathbb{R}^L$, where we assume that this transformation does not affect class membership (for small α). The set of all transformed patterns is now a manifold $\mathcal{M}_{x_t} = \left\{x_t(\alpha) \ : \ \alpha \in \mathbb{R}^L\right\} \subset \mathbb{R}^D$ in pattern space. The distance between two patterns can then be defined as the minimum distance between the manifold $\mathcal{M}_{x_t}$ of the pattern x_t and the manifold $\mathcal{M}_\mu$ of a class specific prototype pattern μ. This manifold distance is truly invariant with respect to the regarded transformations. However, the distance calculation between manifolds is a hard non-linear optimization problem in general. The manifolds can be approximated by a *tangent subspace* $\widehat{\mathcal{M}}$. The *tangent vectors* $x_{t,l}$ that span the subspace are the partial derivatives of $x_t(\alpha)$ with respect to the parameters α_l $(l = 1, \ldots, L)$, i.e. $x_{t,l} = \partial x_t(\alpha)/\partial \alpha_l$. Thus, the transformation $x_t(\alpha)$ can be approximated using a Taylor expansion at $\alpha = 0$:

$$x_t(\alpha) = x_t(0) + \sum_{l=1}^{L} \alpha_l x_{t,l} + \sum_{l=1}^{L} \mathcal{O}(\alpha_l^2) \tag{9}$$

The set of points consisting of the linear combinations of the tangent vectors $x_{t,l}$ added to x_t forms the tangent subspace $\widehat{\mathcal{M}}_{x_t}$, a first-order approximation of $\mathcal{M}_{x_t}$:

$$\widehat{\mathcal{M}}_{x_t} = \left\{ x_t + \sum_{l=1}^{L} \alpha_l x_{t,l} \ : \ \alpha \in \mathrm{I\!R}^L \right\} \subset \mathrm{I\!R}^D \tag{10}$$

Using the linear approximation $\widehat{\mathcal{M}}_{x_t}$ has the advantage that distance calculations are equivalent to the solution of linear least square problems, or equivalently, projections into subspaces, which are computationally inexpensive operations. The approximation is valid for small values of α, which nevertheless is sufficient in many applications, as Fig. 4 shows for example of an image frame of BOSTON50 dataset. These examples illustrate the advantage of tangent distance over other distance measures, as the depicted patterns all lie in the same subspace and can therefore be represented by one prototype and the corresponding tangent vectors. The tangent distance between the original image and any of the transformations is therefore zero, while the Euclidean distance is significantly greater than zero. Using the squared Euclidean norm, the tangent distance is defined as:

$$d(x_t, \mu) = \min_{\alpha,\beta \in \mathrm{I\!R}^L} \left\{ ||(x_t + \sum_{l=1}^{L} \alpha_l x_{t,l}) - (\mu + \sum_{l=1}^{L} \beta_l \mu_l)||^2 \right\} \tag{11}$$

This distance measure is also known as two-sided tangent distance. To reduce the effort for determining $d(x_t, \mu)$, it may be convenient to restrict the tangent subspaces to the derivatives of the reference or the observation. The resulting distance measure is then called one-sided tangent distance. In this work, we replaced the Euclidean distance with the one-sided tangent distance using the derivatives of the mean image μ_{s_t} in state s_t.

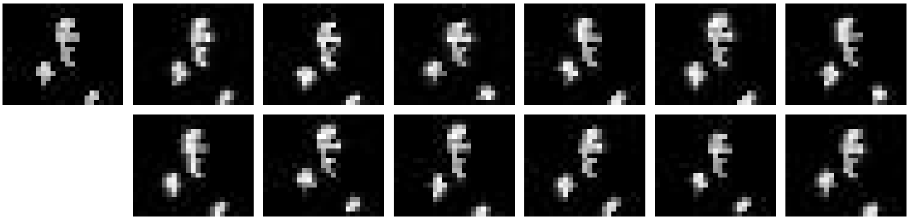

Fig. 4. Example of first-order approximation of affine transformations (Left to right: original image, ± horizontal translation, ± vertical translation, ±axis deformation, ± diagonal deformation, ± scale, ± rotation)

6 Clustering

Due to the high variability of utterances for each word in the database, we consider different pronunciations for utterances of each word. Note that this approach involves a tradeoff; while we may be able to better model the different pronunciations when we use separate HMMs, we are left with fewer data to estimate the HMMs from. We employ and compare three methods of clustering to determine the partitioning into clusters.

Manual Clustering. We observed that there are large visual differences between the utterances of each word, and that they are visually distinguishable. Thus, we are able to label the utterances of different pronunciations for each word as a baseline. We separated the 483 utterances of the BOSTON50 database to 83 pronunciations for the 50 words. The results obtained using this method serve as a lower bound for the automatic methods described in the following because we cannot hope to obtain a better cluster structure. Obviously, for any larger task it will not be feasible to perform a manual labelling. Interestingly, as the experimental results show, the automatic methods can yield error rates that are close to the ones obtained with manually selected labels.

k-means Clustering. One basic but very popular clustering approach is the k-means clustering method. In this method the number of clusters is assumed to be known beforehand and equal to k. We choose one utterance of each of the clusters that were labeled manually as a seed in the initialization. The algorithm continues by adding other utterances to the cluster.

In this algorithm for all words of the database: after initializing k (number of the clusters) and calculating the μ_i as the mean of a the Gaussian function made by utterances of each cluster, all samples would be classified to the nearest cluster. This would be repeated until no change happens in clusters.

LBG-Clustering. The k-means clustering still uses some manually extracted information, i.e. the number of clusters and the initializing seeds of the clusters. We employ the LBG-clustering algorithm proposed by [10] to overcome this constraint and obtain a fully automatic clustering algorithm. This method is described as follows: We perform the clustering for all words of the database as it is shown in Figure 5. First, we assume that all utterances belong to one cluster or particular pronunciation and create an HMM with all utterances existing for a word. If the criterion for dividing a cluster is met, we divide this HMM into two

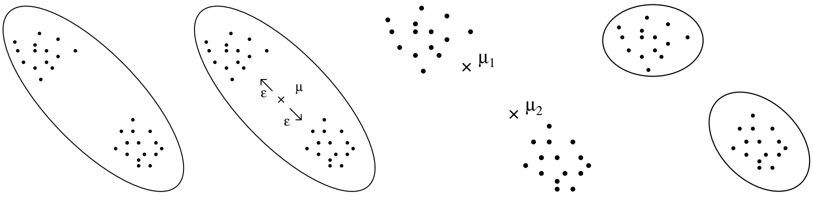

Fig. 5. The LBG-clustering

new cluster centers by adding or subtracting a small value to all means of the states in the model. Then we calculate the similarity between all possible pairs of cluster centers for the word and merge them if the criterion for merging is met. We continue to divide and merge the clusters until no change in the cluster assignment occurs.

The criterion function is defined to calculate the dispersion or scattering of the utterances in a cluster. We use the mean squared distance of the utterances to the mean model as a measure of scatter and normalize that value to the range $[0, 1]$. We consider a threshold value for this criterion function to control the coarseness of the clustering.

Nearest Neighbor Classifier. Nearest neighbor classification is a special case in modelling of the different pronunciations. In nearest neighbor classification the number of pronunciations is considered to be equal to the number of the training utterances for each word. Using each training utterance in the database, we create an HMM. According to the leaving one out method used in this work we separate an utterance as a test utterance from the database. This unknown utterance is classified as belonging to the same class as the most similar or nearest utterance in the training set of the database. This process is repeated for all utterances in the database.

7 Experimental Results

The experiments have been started by employing an HMM for each word of the BOSTON50 database resulting in an error rate of 28.4% with Euclidean distance. We repeated the experiment using the different proposed clustering methods and the tangent distance.

The results are summarized in Table 1. The results show that, in all experiments, tangent distance improves the error rate of the classifiers by between 2 and 10 percent relative. Furthermore, employing clustering methods and the nearest neighbor classifier yields a lower error rate than obtained without considering different pronunciations. The threshold value used in LBG-clustering is a normalized value. When the threshold value is set to 1, no clustering occurs, and when it is set to 0 each utterance will form a separate cluster and the classifier converges to the nearest neighbor classifier. The error rate of the classifier using LBG-clustering with respect to the threshold value is shown in Fig. 6. We

Table 1. Error rates [%] of the HMM classifier with different distances and clusterings

	Euclidean Distance	Tangent Distance
No Clustering	28.4	27.7
Manual Partitioning	22.8	**20.5**
k-means Clustering	23.8	21.3
LBG Clustering	23.2	21.5
Nearest Neighbor	23.6	22.2

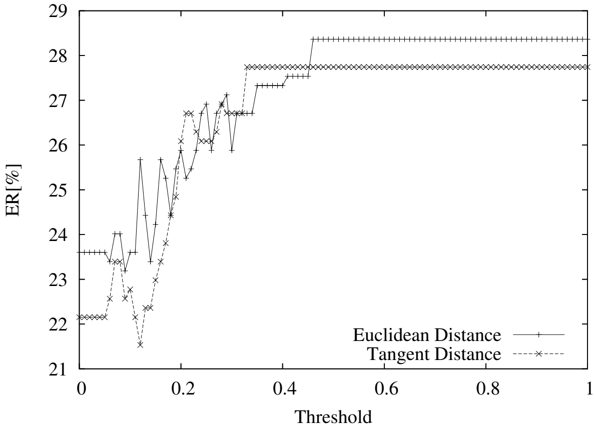

Fig. 6. Error rate of the system with respect to the threshold of clustering

can observe that, with a threshold value of 1, no clustering happens and the error rate is equal to the error rate of the classifier without any pronunciation modeling. When decreasing the threshold value, the error rate is reduced and we can achieve the best error rate of 23.2% and 21.5% using the Euclidean distance and the tangent distance, respectively. The fluctuations we can observe in the diagram for threshold values between 0 and 0.4 lead us to the conclusion that the determination of the best threshold value is not very reliable. Nevertheless, we can observe that there is a strong trend of reducing error rates for smaller threshold values. This leads us to consider the nearest neighbor classifier, which corresponds to the threshold value zero and achieves error rates of 23.6% and 22.2% with the Euclidean distance and the tangent distance, respectively. Because these values are only slightly less than the best –but unstable– result for LBG clustering, this approach should be considered for tasks with a large variability of utterances.

The best error rate of 20.5% is achieved using manual clustering and tangent distance but the results achieved using other clustering methods will be preferable for large databases because they do not involve human labeling of video sequences. The best pronunciation clustering method without human intervention is the hierarchical LBG-clustering with tangent distance and an error rate of 21.5%, which is an improvement of over 22 percent relative.

In the experiments reported above, mixture densities with a maximum number of five densities are used in each state. We have repeated the experiments employing single density and mixture densities, consisting of more densities, in the states of the HMMs. Table 2 shows the results of the experiments employing the tangent distance and different clustering methods. The results show that using a higher number of densities within a mixture density improves the accuracy of the system. In other words, the mixture densities can model the variability of the utterances even without employing the clustering methods. The error rate of the system without any clustering method is 22.8%. In most experiments, the better results are achieved when mixture densities are used in the states. When

Table 2. Error rates [%] of the HMM classifier employing single and mixture densities

	Single Density	Mixture Density
No Clustering	47.4	22.8
Manual Partitioning	35.4	21.9
k-means Clustering	33.1	21.1
LBG Clustering	21.7	22.1

mixture densities are used, the influence of different clustering methods on the error rate of the system is much less than single density experiments.

About half of the remaining errors are due to visual singletons in the dataset, which cannot be classified correctly using the leaving one out approach. This means that one word was uttered in a way that is visually not similar to any of the remaining utterances of that word. For example, all but one of the signs for POSS show a movement of the right hand from the shoulder towards the right side of the signer, while the remaining one shows a movement that is directed towards the center of the body of the signer. This utterance thus cannot be classified correctly without further training material that shows the same movement. This is one of the drawbacks of the small amount of training data available.

A direct comparison to results of other research groups is unfortunately not possible here, because there are no results published on publicly available data so far, and research groups working on sign language or gesture recognition usually use databases that were created within the group. We hope that other groups will produce results for comparison on the BOSTON50 database in the future.

8 Conclusion

In this paper we introduced an appearance-based sign language recognition system. According to our results, considering different pronunciations for sign language words improves the accuracy of the system.

Due to the modeling of different pronunciations of each word in the database, we employed three kinds of the clustering methods; manual clustering, k-means clustering and hierarchical LBG-clustering. These methods can be chosen according to the size of the database in different applications.

Although manual clustering gives more accuracy, it needs manually extracted information and can therefore only be employed for small sets of data. The k-means clustering needs less initial information and only needs to be initialized with the number of clusters and manually selected seed utterances, so this method is also suitable for medium size databases. In contrast, the LBG-clustering method partitions the data automatically and is preferable for large databases where extracting labels manually is unfeasible. According to the results of the experiments on the BOSTON50 database, LBG-clustering leads us to use the nearest neighbor classifier that performs surprisingly well. In all experiments, the tangent distance was compared to the Euclidean distance within the Gaussian emission densities. Using the tangent distance that models small global affine transformations of the images improves the accuracy of the classifier significantly.

References

1. Y. Nam and K. Wohn. Recognition of Space-Time Hand-Gestures Using Hidden Markov Model. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 51–58, Hong Kong, July 1996.
2. B. Bauer, H. Hienz, and K.F. Kraiss. Video-Based Continuous Sign Language Recognition Using Statistical Methods. In *Proceedings of the International Conference on Pattern Recognition*, pp. 463–466, Barcelona, Spain, September 2000.
3. T. Starner, J. Weaver, and A. Pentland. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, December 1998.
4. C. Vogler and D. Metaxas. Adapting Hidden Markov Models for ASL Recognition by Using Three-dimensional Computer Vision Methods. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 156–161. Orlando, FL, October 1997.
5. M. Zahedi, D. Keysers, and H. Ney. Appearance-based Recognition of Words in American Sign Language. *2nd Iberian Conference on Pattern Recognition and Image Analysis*, Volume LNCS 3522 of Lecture Notes in Pattern Recognition and Image Analysis, pp. 511–519, Estoril, Portugal, June 2005.
6. D. Keysers, W. Macherey, and H. Ney. Adaptation in Statistical Pattern Recognition Using Tangent Vectors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):269–274, February 2004.
7. C. Neidle, J. Kegl, D. MacLaughlin, B. Bahan, and R.G. Lee. *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure.* MIT Press, Cambridge, MA, 2000.
8. L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, 77(2):267–296, February 1989.
9. P. Simard, Y. Le Cun, and J. Denker. Efficient Pattern Recognition Using a New Transformation Distance. In *Advances in Neural Information Processing Systems 5*, pp. 50–58, Morgan Kaufmann, 1993.
10. Y. Linde, A. Buzo, and R. Gray. An Algorithm for Vector Quantization Design. *IEEE Trans. on Communications*, Vol. 28, pp. 84–95, January 1980.

Appendix: Visual Lexicon Data

The BOSTON50 database consists of 50 sign language words that are listed with the number of occurrences here:

IX$_i$ (37), BUY (31), WHO (25), GIVE (24), WHAT (24), BOOK (23), FUTURE (21), CAN (19), CAR (19), GO (19), VISIT (18), LOVE (16), ARRIVE (15), HOUSE (12), IX$_i$“far” (12), POSS (12), SOMETHING/ONE (12), YESTERDAY (12), SHOULD (10), IX-1p (8), WOMAN (8), BOX (7), FINISH (7), NEW (7), NOT (7), HAVE (6), LIKE (6), BLAME (6), BREAK-DOWN (5), PREFER (5), READ (4),COAT (3), CORN (3), LEAVE (3), MAN (3), PEOPLE (3),THINK (3), VEGETABLE (3) VIDEOTAPE (3), BROTHER (2), CANDY (2), FRIEND (2), GROUP (2), HOMEWORK (2), KNOW (2),LEG (2), MOVIE (2), STUDENT (2), TOY (2), WRITE (2).