

Automatic Sign Segmentation from Continuous Signing via Multiple Sequence Alignment

Pinar Santemiz¹, Oya Aran², Murat Saraclar³, Lale Akarun¹

¹Department of Computer Engineering, Bogazici University
Istanbul, Turkey

{pinar.santemiz, akarun}@boun.edu.tr

²Idiap Research Institute
Switzerland

Oya.Aran@idiap.ch

³Department of Electrical and Electronics Engineering, Bogazici University
Istanbul, Turkey

murat.saraclar@boun.edu.tr

Abstract

In order to build a sign language recognition framework, one needs to collect sign databases that contain multiple samples of isolated signs, which is a hard and time consuming task. In this study, our aim is to obtain such a database by automatically extracting isolated signs from continuous signing, recorded from the broadcast news for the hearing-impaired. We present an unsupervised, multiple alignment-based approach for sign segmentation. Among the modalities used to form a sign, hand gestures carry most of the information, manifested as hand motion and shape. To handle these two sources of information, we experimented with different feature sets, with different fusion methods on different alignment approaches: feature concatenation on Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs), modeling via coupled and parallel HMMs, and sequential fusion of DTW and HMM. Our experiments on Turkish broadcast news videos show that (1) using low level shape descriptors is suitable for the alignment task, (2) the highest accuracy is obtained by modeling the signs with HMM using the intervals found previously by DTW.

1. Introduction

Sign languages are the natural communication media of the hearing-impaired. They are visual languages which make use of multiple modalities such as hand gestures, body movements and facial expressions. These modalities are expressed together to form a sign. Signs correspond to

words in spoken languages. When expressed in a continuous sequence to form sentences, co-articulation effects are observed, making segmentation a challenging task.

In this paper, we describe a novel approach to automatically extract isolated signs from continuous signing in order to generate usable data for sign language recognition, sign language education and automatic sign language dictionary extraction. In the sign language dictionary Signiary [2], the user enters a word as text and receives the video of the corresponding sign. Our sign video source is the broadcast news videos for the hearing impaired recorded from the Turkish Radio-Television (TRT) channel, which contain the video of the news presenter simultaneously signing and speaking. The signing in these news videos is considered as “signed Turkish”, in which the sign of each word is from Turkish Sign Language (TSL) but their ordering would have been different in a proper TSL sentence. In the sign language dictionary Signiary [2], the speech intervals for the queried word are broadened; producing a short video clip that contains the desired sign, with parts of the preceding and following signs. However, the exact beginning and end locations of the sign can be anywhere in the interval, since speech and sign are weakly synchronized. Since the queried word can be articulated many times within the news videos, we have many such video clips, whose common property is that they contain the same sign, with differing beginning and end parts. We define our problem as “finding the longest common subsequence in multiple sequences” and use alignment techniques to solve this problem. Figure 1 shows the flowchart of our approach. When the user queries

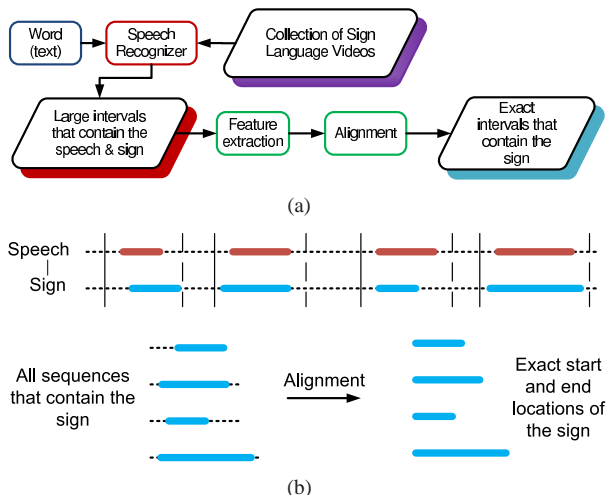


Figure 1. System flowchart. (a) General system flow, (b) Red lines show the places where the word is found in the speech and blue lines show the actual, ground truth, places where the sign of the word is performed. Given the intervals around the red lines, we aim to find the locations of the blue lines.

a word by entering text, the speech intervals that contain the word in the speech modality are extracted [2]. In this work, we focus on the sign alignment problem and assume that the speech recognition is already performed [19], and speech intervals given as input to our system.

Research on sign language analysis aims to realize several applications that are based on sign language recognition, sign segmentation, or sign clustering. For describing the temporal characteristics of the signs, several methods are used such as DTW [9], HMMs [12], or extensions of HMMs [4, 21, 1]. Some isolated gesture recognition systems use DTW based methods to align and compare temporal characteristics of the gestures [9]. On the other hand, in the continuous gesture recognition task, HMMs are preferred due to their ability to implicitly segment continuous sequences [12]. In [1], both HMM and DTW are used to jointly segment and recognize continuous signing. An alternative method is to model the transition parts, rather than the signs, for sign segmentation and recognition [12].

Sign language consists of multiple interacting processes that complement each other to convey information. To model these processes, while preserving their natural correlation, extensions on HMMs such as coupled HMMs [4] and parallel HMMs [21] have been proposed. In [4], coupled HMMs are used for upper-body gesture recognition where each arm is treated as a process. In [21], parallel HMMs are used to model 3D data of continuous American Sign Language (ASL), where left and right hand are modeled independently using HMMs and their probabilities are later combined at word or phoneme ends.

Sign segmentation is generally used as a subtask, and

handled implicitly as in [1, 21]. In [13], a discriminative method for sign spotting in English and ASL is presented. Here, a rough alignment via HMMs is followed by sign spotting based on a discriminative model. Recently, there has been an interest on using speech or subtitles to extract isolated signs from continuous signing and three papers with similar objectives are presented at the same conference [7, 5, 16]. In [7] data mining methods are used to segment the signs, where the segmentation is guided by subtitles. In [5], subtitles are used to define broad locations of signs and multiple instance learning is used to decide whether an interval contains the sign. In [16], iterative conditional modes method is used to extract signemes, which is defined as the parts of signs that is common in all instances.

Our contributions in this study are: (1) We present a method to automatically extract and segment isolated signs from continuous sentences in an unsupervised way via multiple sequence alignment techniques and propose a technique based on the sequential fusion of DTW and HMM, (2) We use available TV sources, such as the broadcast news for the hearing-impaired, to collect a sign database; (3) We apply and adapt different alignment algorithms and fusion techniques to sign segmentation problem and present a detailed analysis and comparison of different features of hand motion and shape on the alignment performance. As the newscasters in our videos are native signers and the signing takes place in an unconstrained environment at high speed, our proposed system is suitable to work on natural signing.

The paper is organized as follows: The details of the database are given in Section 2. In Section 3 we explain our particle filter based tracking algorithm. Feature extraction techniques are summarized in Section 4. The multiple sequence alignment techniques are explained in Section 5. Experimental results and conclusions are given in Sections 6 and 7, respectively.

2. Database

We use a database of 15 video recordings of TRT broadcast news for the hearing impaired. In all of the videos, the same newscaster is presenting the news by speaking and signing simultaneously. The total length of the videos is around two hours, with 174939 frames and a total of 10318 words. These words correspond to 3498 different signs. The exact start and end locations of the signs are manually annotated by TSL signers. A sample of 40 words, among the most frequent ones, are selected from the whole database, where each word has 30 samples. For these 40 words, the accuracy of the speech recognizer is 100%, the average sign duration with respect to manual annotation is 15.72 frames and the average duration of the corresponding speech interval is 15.99 frames.

Among these 40 signs, half of them are one handed, performed with the right hand, and the other half is two handed.

Further analysis about the 40 selected signs can be done with respect to the occlusions and contact of the hands and the face. The number of signs in which there is an occlusion or contact of the two hands or the hand and the face is 15 and 7, respectively. Having more than half of the signs with occlusion or contact, we can describe our database as a challenging one as dealing with occlusions and contacts is difficult in tracking, feature extraction and alignment.

3. Tracking

Tracking hands and face without markers during signing is a challenging task due to the occlusions and interactions of the hands and the face. Moreover, the hands move quite fast and sometimes cross (i.e. left hand is on the right of the right hand or vice versa), making it unrealistic to make assumptions for the relative locations of the hands. A tracking algorithm that aims to perform markerless hand tracking in natural speed signing should be robust to occlusions and contacts, fast hand speed, and hand crossing.

In this work, we use a joint Particle Filter (PF) based method that can robustly track the hands and the face during natural signing. We use a joint PF to track a maximum of three objects: two hands and the face. The complexity of the joint PF is reduced by embedding mean shift tracking, which allows us to achieve similar tracking accuracy by using substantially fewer particles. We handle the occlusions by updating the likelihood of the particles with respect to their proximity and forcing them to be as separate as possible. The method is robust to occlusions and is able to recover fast if the tracking fails. Figure 2 shows a sample frame and the particle distributions with the joint PF. Details of the tracking algorithm can be found in [6].

To evaluate the performance of tracking, we manually annotated the videos in the database, creating a ground truth for the center of mass coordinates of the hands and the face. With this ground truth data, we followed two evaluation approaches: frame based and sample based. In the frame based evaluation, for each frame, we compared the ground truth with the found positions. If the distance between the two positions is less than the length of the shorter axis of the found ellipse we assume that the tracking is correct. We evaluate the performance of tracking on the sign videos in the database. With this approach, we achieved a tracking accuracy of 97% for the hands and around 99% for the face. For the sample based evaluation, we considered each sign video as a sample and denoted the tracking accuracy of that sample as erroneous, when the tracking error continues for more than three frames. Consequently, the sample based tracking accuracy on our database is found to be 98.6% for the hands, and 100% for the face.

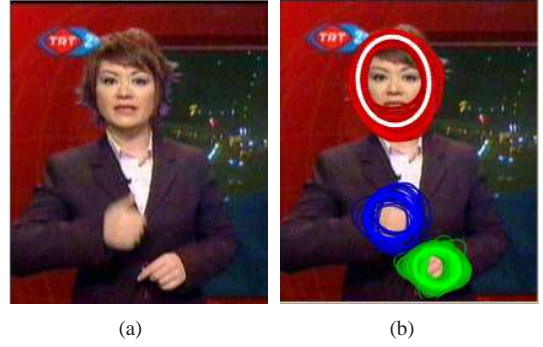


Figure 2. Tracking with joint particle filter: (a) Original image, (b) Particle distribution in joint particle filter.

4. Feature Extraction

We use the output of the tracking algorithm, which consists of the center of mass coordinates, and the bounding ellipse parameters for each hand and the head. However, to extract accurate hand shape information, we need to find a segmented image containing the whole hand. For this purpose, we apply region growing for each hand starting from the skin color region enclosed by the bounding ellipse. The resulting images may be noisy due to occlusion or image characteristics. Therefore, after region growing, we apply template matching and obtain the segmented hand images.

4.1. Features of hand motion and shape

We extract seven feature sets as illustrated in Figure 3: 1) Center of mass coordinates of the hands (\mathbf{C}) and 2) their first order derivatives ($\Delta\mathbf{C}$), 3) Ellipse parameters for each hand: major and minor axes and rotation angle (\mathbf{E}), 4) Radial distance function descriptors (\mathbf{R}), 5) Hu moments (\mathbf{Hu}), 6) Discrete cosine transform coefficients (\mathbf{D}), and 7) Histogram of oriented gradients (\mathbf{H}).

4.2. Center of Mass (\mathbf{C}) and Ellipse Parameters (\mathbf{E})

As a simple shape descriptor to represent the hand shape, we fit an ellipse to the segmented hand images and calculate the ellipse parameters: center of mass coordinates, major and minor axes and the rotation angle.

Sequences representing the trajectory may contain gaps when the hands disappear. Hence, we first fill the gaps using linear interpolation. To obtain translation invariance, we take the face as the center of our coordinate system and recalculate the center of mass coordinates of the hands accordingly. Then, we apply a moving average filter to the features along the hand trajectory to eliminate noise. Finally, we normalize the coordinates between 0 and 1 using min-max normalization to obtain scale invariance. The feature vector sets consist of center positions (\mathbf{C}), their first

order derivatives (ΔC), and ellipse parameters (E) for left and right hands.

4.3. Radial Distance Function (R)

Radial Distance Function (RDF) is a method to describe the outer contour of an image. In hand gesture recognition, when fingers are visible and separated, RDF can be used to detect and localize the fingertips [15]. To obtain RDF features, a reference point inside a closed contour is chosen and the distance of this reference point to the curve as a function of angle is plotted. In our computations, we take the center of mass coordinate as the reference point and compute R for every five degrees.

4.4. Hu Moments (Hu)

Image moments are used in several computer vision applications for representing global and invariant shape characteristics of image features. In our study we use seven Hu moments obtained from the binary mask of the segmented hand image. These moments are scale, translation, rotation invariant pattern identification. The first six are also reflection invariant whereas the seventh moment is skew orthogonal invariant, which is useful in distinguishing mirror images. The equations can be found in [14].

We observed a small improvement in the performance when we included the rotation information to our feature vector. Therefore, prior to calculating the Hu moments, we rotate the images using the rotation angle that we obtained from the computation of ellipse parameters, and we attach the rotation angle to our final feature vector.

4.5. Discrete Cosine Transform (D) coefficients

Discrete cosine transform (DCT) is used for data representation and classification in many applications such as face [10, 11] and object recognition [3]. We use DCT on hand images to extract features to represent the hand shape.

Prior to calculating the DCT coefficients, we performed some preprocessing on the hand images. First, we compensated for the rotation on the hand, and crop the image to 64×64 pixels size with the hand located in the center. Then, we fill the background with the mean color of the segmented area and convert the image to gray scale.

We divide each hand image into blocks of size 8×8 pixels and on each block we apply DCT. As most of the information in DCT is concentrated in the lower frequencies, we order the DCT coefficients using zig-zag scanning, eliminate the DC coefficient and take the first five coefficients in each of the blocks. For normalization, we use a similar approach as in [11]. To eliminate the effect of illumination changes in each block, we normalize the total magnitude of each block's DCT coefficients to unit norm. Then, to balance the effect of each coefficient, we divide the coeffi-

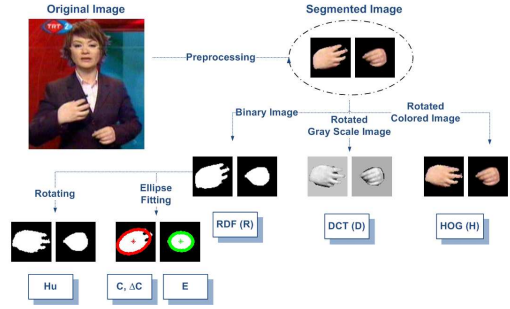


Figure 3. Feature extraction techniques

icients to the standard deviations learned from a training set. Finally, we reduce the dimensionality with Principal Component Analysis (PCA) to obtain vectors with 11 features for each hand, which explains 90 percent of the variance.

4.6. Histogram of Oriented Gradients (H)

Histogram of Oriented Gradients (HOG) are mainly used in computer vision as feature descriptors in object detection and recognition [8]. HOG represents the shape via the distributions of local intensity gradients or edge directions. The main advantage of using HOG descriptors is that they offer some robustness to scene illumination changes, while capturing characteristic edge or gradient structure.

As a preprocessing step, we rotate the hand using the rotation angle and translate it to the center of a 64×64 pixel box. For each color channel, we compute the gradient image using the centered 1D point derivative, which is the mask $[-1, 0, 1]$, both in the vertical and horizontal directions. Then we compute the gradient magnitude and orientation, and choose the values having the largest norm for the magnitude as the corresponding pixel's values.

We divide the image into non-overlapping cells with 8×8 pixels. For each cell, we form an orientation histogram having nine bins, evenly spaced between $[-\pi/2, \pi/2]$. Each pixel in the cell calculates a weighted vote for the histogram based on the gradient orientations. For normalization, we group cells into overlapping blocks of 16×16 pixels and extract feature vectors, with magnitude normalized to unit norm. As a final step, we reduce the dimensionality using PCA so that 90% of the variance is explained, and obtain vectors with 11 features for each hand.

5. Alignment

Since speech and signing are not fully synchronized, in our database, the speech information starts and ends approximately 7 frames later than the signing. Therefore, we enlarge the speech intervals by 15 frames in the beginning and shorten them by one frame in the end to guarantee that more than 90% of the signs are included by the intervals. We used

four different alignment techniques: DTW, HMM, coupled HMM, and parallel HMM.

5.1. Dynamic Time Warping (DTW)

DTW is a widely used alignment approach in the field of bioinformatics [17] and speech recognition [18]. Generally, it is used for pairwise alignment since the extension of the problem is NP-complete. In this study, we perform the multiple alignment of the sequences via pairwise alignments.

First, we calculate the local score matrix of the two sequences using Euclidean distance as local distance. Each element of the matrix corresponds to the distance between the feature vectors at corresponding frames. Once the score matrix is calculated we need to find the alignment path satisfying the following conditions: the path starts and ends in diagonally opposite corner cells of the matrix, it must be continuous and monotonically spaced in time. To satisfy these conditions, an accumulated distance matrix D_A is constructed from the local distance matrix, D_S using:

$$D_A(i, j) = D_S(i, j) + \begin{cases} D_A(i, j-1), & i = 1, \\ D_A(i-1, j), & j = 1, \\ \min \begin{pmatrix} D_A(i, j-1), \\ D_A(i-1, j), \\ D_A(i-1, j-1) \end{pmatrix}, & i, j > 1 \end{cases} \quad (1)$$

When the accumulated distance matrix is found, it is traversed backwards by choosing the minimum elements to form the pairwise alignment path. In our case we know that there is a high possibility of having junk frames at the start and end of the sequences. Therefore, unlike the general approach, we start forming the accumulated distance matrix from an interior position to increase the possibility of starting from a frame belonging to the sign. Our algorithm determines a window having one third the dimensions of this matrix and located in the center, and select the location with minimum local distance inside this window as starting position. From this point we move forward and backward to find the corresponding alignment path.

The start and end locations of the sign are determined by analyzing the local scores along the alignment path. One can assume that local scores decrease when the sign starts and increase when the sign ends. Hence, the start and end locations correspond to the local maxima points of the local scores along the alignment path (see Figure 4). As a result of the pairwise alignments, we obtain candidate start and end locations for each pairwise alignment. The final locations are determined by averaging the candidate locations.

5.2. Hidden Markov Models (HMM) and variants

HMMs are generative probability models that provide an efficient way of dealing with variable length sequences and missing data [20]. HMMs draw much attention with their

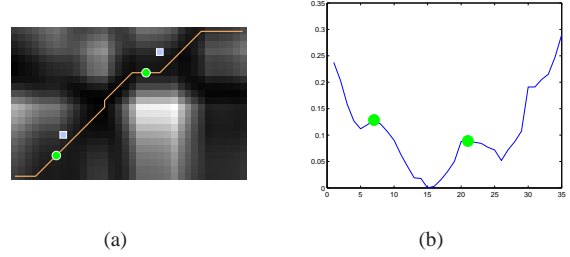


Figure 4. Alignment with DTW. (a) The local score matrix of two videos and the alignment path. Green circles show the found locations, blue squares show the ground truth locations. (b) The changes in the local score on the alignment path.

ability to cope with the temporal variability among different instances of the same sign. Left-to-right HMMs are preferred for their simplicity and suitability to sign modeling.

In this work, we use two modalities, hand motion and shape, represented by a set of features. These modalities must be combined according to their synchronization with each other. The simplest solution is to put the features of all the concurrent modalities in a single feature vector, which assumes that the modalities are in full synchronization. Instead of concatenating the features into a single feature vector, a model can be dedicated for each modality with established links between the states of different processes. Coupled HMMs have been proposed for coupling and training HMMs that represent different processes with loose synchronization [4]. When the synchronization of the modalities is very weak, parallel HMMs can be used [21].

We model each sequence with a left-to-right model with a state number proportional to the number of frames in the sequence. We assume that in each sequence, there are junk frames unrelated to the sign at the start and end of the sequence. Our aim is to find the part that contains only the sign of the word, using HMMs, coupled HMMs or parallel HMMs. In each case, the training is performed by leave-one-out cross validation, with one example in the test set and the remaining examples in the training set. The start and end frames are selected as follows: For each sign, the frame where the first state ends is taken as the start frame and the frame where the last state starts is taken as the end frame. In case of parallel and coupled HMMs, both of the processes are taken into account and for each sign, the frame where the first state ends in both of the processes is taken as the start frame and the frame where the last state starts in both of the processes is taken as the end frame.

HMM: We train a single Gaussian HMM using Baum-Welch algorithm for each word, such that the start and end states are common for all the sequences (see Figure 5(a)).

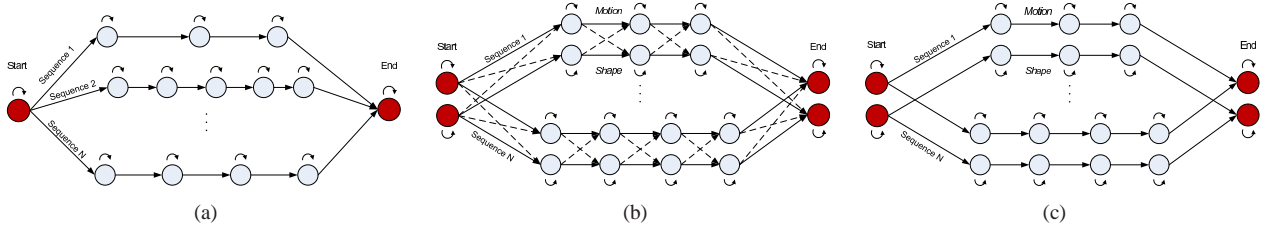


Figure 5. Alignment with (a) HMM, (b) Coupled HMM, (c) Parallel HMM.

Parallel HMM: We train two independent and parallel HMMs for each word where the motion and shape modalities are modeled as different processes. The training is done in a similar approach as in HMMs (see Figure 5(c)).

Coupled HMM: We train a coupled HMM for each word, in which the motion and shape are modeled as different processes and coupled to model the loose synchronization in between. The junk frames at the start and end are considered for each modality and coupled as the rest of the sequence. The training is done such that the start and end states are common for all the sequences (see Figure 5(b)).

Sequential fusion of DTW and HMM: We combine HMM and DTW by training HMMs on the intervals that are found by the DTW algorithm. When the interval including the sign is wide, a high number of observations correspond to junk states while the inner states are expected to represent relatively short sequences, which decreases the performance. In our experiments we have seen that DTW can serve for narrowing the search window. Therefore, we first apply DTW to the sequences to reduce the interval and then apply HMM to find the segmentation.

6. Experiments

We used three different performance measures to evaluate the system performance: accuracy, precision, and recall. We calculate these measures by comparing the location of the sign found by the algorithm with the ground truth location via the True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values (Figure 6). Equations 2 - 4 show the calculation of these measures.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

In Table 1, we analyze the performance of shape features and their combinations with motion features on

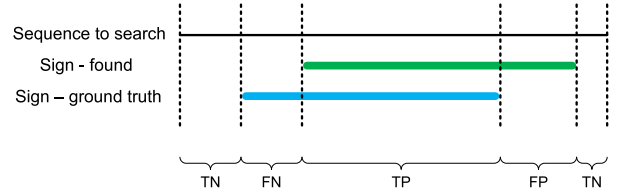


Figure 6. The True Positive (TP), True Negative (TN), False positive (FP) and False Negative (FN) values for the extracted sign with respect to the interval it is searched and the ground truth.

DTW, HMM, and sequential fusion of DTW and HMM (DTW+HMM). When we compare the results for DTW and HMM, we see that DTW consistently perform better than HMM on accuracy and precision. We see that HMM has higher recall rates but less precision. This indicates that the segmentation of HMM covers almost the entire interval. Figures 7(a) and 7(b) show the alignment on a sample word. As can be seen from these alignments, DTW finds successful segmentations unless the synchrony between the speech and the sign is very weak, where as HMMs perform much better in those cases (e.g. fourth sequence of the sample word). To combine the powers of both methods, we apply HMM to the alignments obtained by DTW. When there is better synchrony, alignment of HMM does not significantly change the alignment of DTW. However, in the case of weak synchronization, HMM corrects DTW. Accuracy and precision rates of DTW+HMM are slightly better than that of DTW, with a small compensation in the recall rates. Alignment results on a sample word can be seen in Figure 7(c). The best accuracy is obtained as 80.5%, on the feature set $\Delta C, C, E$ with sequential fusion of DTW and HMM.

The results in Table 2 show that although parallel and coupled HMMs are superior to HMM, they can not beat DTW. Coupled HMM is slightly better than parallel HMMs, which is expected since coupled HMMs model the internal dependencies between the two processes, in our case the two modalities. As in the results of DTW and HMM, the same feature set, $\Delta C, C, E$, gives the best performance.

Among the shape features, the performance of ellipse features are consistently better than other high level shape descriptors. We think that this is due to the low resolution

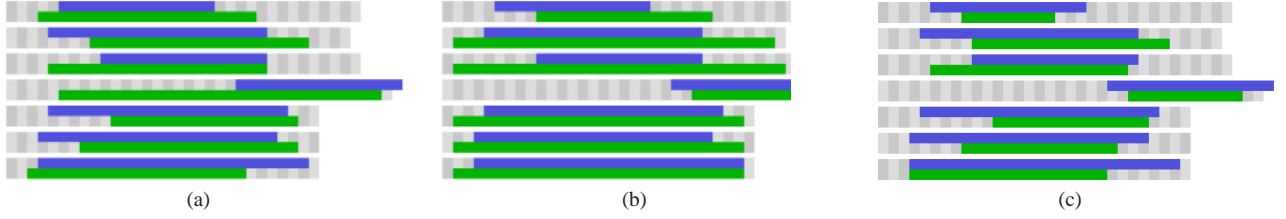


Figure 7. Examples for the alignment of the word “prime minister” for (a) DTW, (b) HMM, and (c) sequential fusion of DTW and HMM, using feature set C, Δ C, E. Each box stands for one frame. The sign is searched within the gray area, the green lines represent the found segment, and the blue lines represent the ground truth.

of the hand shapes and also due to the occlusions with the face and the other hand. Simple shape descriptors such as ellipse features are more robust to such occlusions.

Table 1. Performance of DTW, HMM and DTW+HMM with respect to accuracy, precision, recall.

Feature sets	DTW (%)			HMM (%)			DTW+HMM (%)		
	Acc	Pre	Rec	Acc	Pre	Rec	Acc	Pre	Rec
E	77.98	78.09	85.21	69.36	68.36	82.78	76.59	86.24	68.35
R	75.16	75.56	85.83	65.13	62.59	93.44	75.77	80.84	76.31
Hu	76.61	78.73	80.16	59.89	60.54	76.43	72.68	84.53	61.44
D	75.37	73.52	87.15	56.53	55.39	74.57	70.19	74.89	64.24
H	75.79	74.68	86.01	56.80	56.50	74.42	70.52	77.02	63.20
C,E	79.07	78.84	86.78	72.82	70.88	90.09	80.05	88.33	74.23
C,R	77.12	78.15	85.29	68.76	65.13	95.64	78.43	84.77	76.70
C,Hu	78.52	78.41	86.15	70.81	70.12	88.55	78.90	87.16	73.43
C,D	78.28	76.45	88.81	58.48	61.19	77.10	74.18	79.26	69.53
C,H	78.04	76.54	88.36	57.66	59.23	75.68	74.17	80.23	68.96
Δ C,C,E	79.51	80.18	85.79	72.75	69.66	93.12	80.45	88.24	75.42
Δ C,C,R	77.05	78.29	85.16	68.00	64.37	95.86	78.41	84.45	76.94
Δ C,C,Hu	79.02	81.02	83.58	70.97	69.42	90.52	79.01	88.21	72.46
Δ C,C,D	79.25	79.47	86.33	58.02	59.43	78.36	74.05	80.01	66.30
Δ C,C,H	78.83	79.01	86.24	58.13	59.47	76.73	72.76	79.12	63.94
Δ C,C,E,R	77.90	78.60	86.05	69.63	66.19	95.28	79.51	85.62	77.38
Δ C,C,E,Hu	78.74	79.21	85.70	73.16	71.71	90.31	79.32	87.17	73.89
Δ C,C,E,D	78.81	78.51	86.93	58.68	59.32	77.41	73.40	78.28	65.93
Δ C,C,E,H	78.61	78.01	87.47	59.08	59.70	77.85	64.83	64.68	46.27

Table 2. Performance of coupled and parallel HMMs with respect to accuracy, precision, recall.

Feature sets		Coupled HMM (%)			Parallel HMM (%)		
HMM 1	HMM 2	Acc	Pre	Rec	Acc	Pre	Rec
C	E	74.93	75.94	76.50	73.35	74.08	79.10
C	R	74.83	73.65	84.60	73.33	71.75	88.43
C	Hu	71.52	71.94	73.18	67.37	67.35	72.72
C	D	66.95	64.45	70.23	65.45	62.80	70.73
C	H	66.61	64.61	68.98	65.72	63.93	70.60
C, Δ C	E	75.11	75.81	79.35	73.49	73.94	79.76
C, Δ C	R	73.57	71.42	88.02	73.24	71.28	89.58
C, Δ C	Hu	70.84	71.23	73.19	67.50	67.49	73.39
C, Δ C	D	65.48	62.69	71.00	65.26	62.56	71.23
C, Δ C	H	65.52	62.41	70.71	65.55	64.19	71.28
C, Δ C,E	R	73.84	71.89	86.92	73.99	72.30	88.36
C, Δ C,E	Hu	71.88	72.15	74.45	68.11	67.88	72.75
C, Δ C,E	D	67.10	64.82	70.89	66.25	63.51	70.47
C, Δ C,E	H	66.55	64.49	70.75	66.52	64.85	70.62

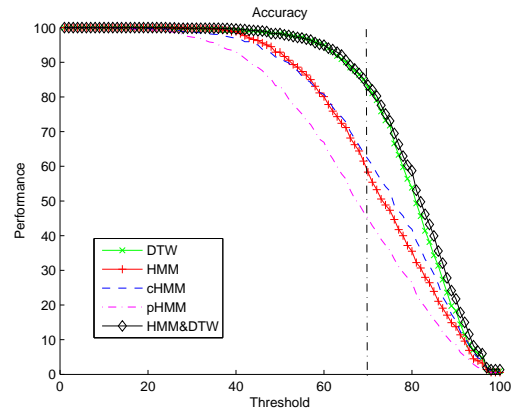


Figure 8. Accuracy of feature set C,E for DTW, HMM, DTW+HMM, coupled HMM, parallel HMM with respect to different correctness thresholds

Exact match with the ground truth is rarely possible due to the uncertainty of the sign boundaries. In many of the works on sign alignment [1],[5], if the overlap with the ground truth is more than 50%, the result is accepted as successful, which we believe that 50% is a low threshold to give such a decision. In some other works, the results are evaluated by sign language experts [16]. To measure the correctness percentage, we set a correctness threshold and assume that the detection is successful if the measures are higher than this threshold. In Figure 8, we show the behavior of the system when we change the correctness threshold. Tables 3(a) and 3(b) show the results for correctness threshold 70%. We observe that the best performance is obtained with sequential fusion of DTW and HMM with an accuracy rate of 83.42%. Note that for correctness threshold 50%, our performance is above 95%.

7. Conclusions

With intensified interest in automatic sign language recognition, the automatic extraction of sign databases has

Table 3. Accuracies of (a) DTW, HMM, DTW+HMM, and (b) Coupled HMM, Parallel HMM with correctness threshold 70

(a)			(b)				
Feature sets	DTW	HMM	DTW+HMM	Feature sets		Coupled	Parallel
	(%)	(%)	(%)	HMM 1	HMM 2		
E	77.17	52.00	71.58	C	E	65.42	63.00
R	67.17	37.75	70.17	C	R	66.50	60.75
Hu	74.17	26.08	62.08	C	Hu	57.33	45.75
D	69.92	17.58	56.83	C	D	46.17	41.67
H	70.17	17.42	57.00	C	H	45.00	41.92
C,E	80.67	59.50	82.75	C, Δ C	E	67.08	62.08
C,R	72.67	47.00	75.92	C, Δ C	R	61.92	60.00
C,Hu	79.50	55.25	78.58	C, Δ C	Hu	56.83	45.67
C,D	78.50	20.92	68.00	C, Δ C	D	41.67	40.58
C,H	77.50	18.92	67.75	C, Δ C	H	42.25	40.25
Δ C,C,E	82.83	58.33	83.42	C, Δ C,E	R	64.58	60.83
Δ C,C,R	72.67	44.67	76.67	C, Δ C,E	Hu	62.25	62.67
Δ C,C,Hu	81.08	54.42	78.33	C, Δ C,E	D	58.92	48.75
Δ C,C,D	81.67	19.33	67.67	C, Δ C,E	H	46.83	43.92
Δ C,C,H	79.58	19.17	64.83				
Δ C,C,E,R	76.67	49.08	79.42				
Δ C,C,E,Hu	79.50	60.58	79.67				
Δ C,C,E,D	79.67	22.08	67.83				
Δ C,C,E,H	77.92	22.25	65.25				

become more important. We use the broadcast news for the hearing-impaired, to automatically extract signs, with the help of speech that coexists with the signs. The automatic extraction of speech intervals enables the unsupervised extraction of isolated signs from continuous signing.

In this paper, we use a multiple alignment based approach: we align the sequences obtained via speech recognition to extract the exact locations of the sign they contain. The received sequences contain the different performances of sign that we search for, however these are broad intervals and roughly contain the sign. We aim to find the longest common subsequence in these multiple sequences, which gives us the sign. Our experiments show that the sequential fusion of DTW and HMM combines the powers of each method successfully and gives the highest accuracy.

8. Acknowledgement

This work is supported by Tübitak-RFBR joint project 108E113 and EU FP7 Marie Curie IEF project NOVICOM.

References

- [1] J. Alon, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Trans. PAMI*, 99(1), 2008.
- [2] O. Aran, I. Ari, P. Campr, E. Dikici, M. Hruz, S. Parlak, L. Akarun, and M. Saraclar. Speech and sliding text aided sign retrieval from hearing impaired sign news videos. *Journal on Multimodal User Interfaces*, 2(1):117–131, 2008.
- [3] N. D. Binh, E. Shuichi, and T. Ejima. Real-time hand tracking and gesture recognition system. In *Proc. of Int. Conf. on Graphics, Vision and Image Processing*, pages 362–368, 2005.

- [4] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. IEEE Conf. CVPR*, page 994, 1997.
- [5] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching tv (using weakly aligned subtitles). In *Proc. IEEE Conf. CVPR*, 2009.
- [6] P. Campr, M. Hruz, A. Karpov, P. Santemiz, M. Zelezny, and O. Aran. Sign language enabled information kiosk. In *4th International Summer Workshop on Multimodal Interfaces (eINTERFACE), Paris, France*, pages 24–33, 2008.
- [7] H. Cooper and R. Bowden. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In *Proc. IEEE Conf. CVPR*, 2009.
- [8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. CVPR*, volume 2, pages 886–893, 2005.
- [9] T. Darrell, I. A. Essa, and A. P. Pentland. Task-specific gesture analysis in real-time using interpolated views. *IEEE Trans. PAMI*, 18(12):1236–1242, 1996.
- [10] H. K. Ekenel and R. Stiefelhagen. Local appearance based face recognition using discrete cosine transform. In *13th European Signal Processing Conf., Antalya, Turkey*, 2005.
- [11] H. K. Ekenel and R. Stiefelhagen. Analysis of local appearance-based face recognition: Effects of feature selection and feature normalization. In *Proc. IEEE Conf. CVPR Workshop*, 2006.
- [12] G. Fang, W. Geo, and D. Zhao. Large-vocabulary continuous sign language recognition based on transition-movement models. *IEEE Trans. on Systems, Man and Cybernetics - Part A: Systems and Humans*, 37(1):1–9, 2007.
- [13] A. Farhadi and D. Forsyth. Aligning asl for statistical translation using a discriminative word model. In *Proc. IEEE Conf. CVPR*, volume 2, pages 1471–1476, 2006.
- [14] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE Trans. on Information Theory*, 8(2):179–187, 1962.
- [15] E. Konukoglu, E. Yoruk, J. Darbon, and B. Sankur. Shape-based hand recognition. *IEEE Trans. on Image Processing*, 15(7):1803–1815, 2006.
- [16] S. Nayak, S. Sarkar, and B. Loeding. Automated extraction of signs from continuous sign language sentences using iterated conditional modes. In *Proc. IEEE Conf. CVPR*, 2009.
- [17] C. Notredame. Recent progresses in multiple sequence alignment: a survey. *Pharmacogenomics*, 3(1):131–144, 2002.
- [18] A. S. Park and J. R. Glass. Unsupervised pattern discovery in speech. *IEEE Trans. on Audio, Speech, and Language Processing*, 16(1):186–197, January 2008.
- [19] S. Parlak and M. Saraclar. Spoken term detection for Turkish broadcast news. In *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 2008.
- [20] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. of the IEEE*, volume 77 of no. 2, pages 257–285, 1989.
- [21] C. Vogler and D. Metaxas. Parallel hidden Markov models for American sign language recognition. In *Int. Conf. on Computer Vision*, pages 116–122, 1999.