# Extraction of Isolated Signs from Sign Language Videos via Multiple Sequence Alignment

*Pinar Santemiz, Oya Aran, Murat Saraclar, Lale Akarun,*

Bogazici University
Istanbul, Turkey
{pinar.santemiz, aranoya, murat.saraclar, akarun}@boun.edu.tr

## Abstract

In this work, we present an alignment-based method to perform sign segmentation and to extract isolated signs from continuous sign language videos. Sign videos contain many modalities, the most prominent of which are hand gestures, manifested as hand motion and shape, which are represented by a variety of extracted features in this work. We compare two different alignment approaches, Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs), and analyze their behaviour with different feature sets on a database from Turkish sign language. Our experiments show that simple hand shape descriptors perform better than the high level ones and the accuracy of HMMs is better than DTW.

## 1. Introduction

Sign languages are the primary means of communication for the hearing-impaired. They are visual languages and make use of multiple modalities such as hand gestures, body movements and facial expressions to convey information. These modalities are expressed in parallel to form a sign. Signs represent concepts: They are counterparts of words in spoken languages. When expressed one after one in a sign language sentence, co-articulation effects are observed, making segmentation a challenging task.

In this study, we aim to extract isolated signs from continuous signing in order to generate usable data for sign language education and automatic sign language dictionary extraction, where the user enters a word as text and receives the video of the sign for the word that is searched [2]. For this purpose, we use an excellent source from the news videos of the Turkish Radio-Television (TRT) channel. The broadcast news is for the hearing impaired and consists of three major information sources: sliding text, speech and sign. The three sources in the video convey the same information via different modalities. The news presenter signs the words as she talks. Moreover, a corresponding sliding text is superimposed on the video. Note that, since it is not necessary to have the same word ordering in a Turkish spoken sentence and in a Turkish sign sentence [15], the signing in these news videos is not considered as Turkish Sign Language (TSL) but can be called as "signed Turkish", in which the sign of each word is from TSL but their ordering would have been different in a proper TSL sentence.

As the speech and sliding text are parallel with the signs, one can assume that a sign corresponding to a word can be found around the location where the same word is spoken and exists in the sliding text. This location will give a broad interval for the possible location of the sign but as the speech and sign are not fully synchronized, finding the exact start and end

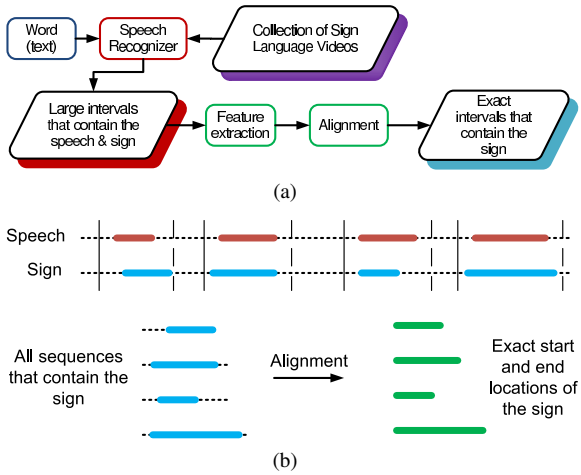



Figure 1: System flowchart. (a) General system flow, (b) Red lines show the places where the word is found in the speech and blue lines show the actual, ground truth, places where the sign of the word is performed. Given the intervals around the red lines, we aim to find the locations of the blue lines.

locations of the sign is a challenging problem. A word can be articulated many times and for each of these, the same sign will be performed. Thus, for a single word, we will have multiple videos that include different performances of the same sign. We define this problem as "finding the longest common subsequence in multiple sequences" and use alignment techniques to solve the problem. Figure 1 shows the flowchart of our approach. The user enters a word as text and the word is searched in the speech [2]. The intervals that contain the word in the speech modality are extracted. These intervals are assumed to contain the sign of the word. By using alignment techniques, we find the longest common subsequence in these intervals, which gives us the exact locations of the signs. Note that the system does not need pre-learned sign models and the extraction is done via alignment techniques on the intervals detected by the speech recognizer. In this work we assume that the speech recognition is already performed [13] and for each word to be searched, the intervals that contain the signs are given as input to our system.

Sign language recognition has drawn much attention in the past 10 years. Regardless of the method used, these systems need a database for the signs they want to recognize and this database is used to train a model that will be used in the recognition task. Although there are studies that use DTW based meth-

ods [6] for isolated recognition, HMMs generally show better performances and are easily applied to continuous recognition tasks. In the continuous case, the success of HMMs comes from the fact that they can implicitly segment continuous sequences. In [1], both HMM and DTW are used to jointly track, segment and recognize continuous signing. Here, continuous image streams with moving, cluttered backgrounds are used, which makes the hand location highly ambiguous. An alternative segmentation method is proposed in [8], where instead of modeling the signs, transition parts are modeled via HMMs.

Although sign segmentation is implicitly handled in continuous recognition tasks [8], direct sign segmentation or spotting is rarely addressed. Recently, in [9], a discriminative method for sign spotting to align English and American Sign Language (ASL) is presented. The authors first apply a rough alignment via HMMs, followed by sign spotting based on a discriminative model. Our contribution in this work is to perform direct sign segmentation via sequence alignment techniques, without the need for a pre-trained sign model. Similar applications can be found in the field of bioinformatics, where several sequence alignment methods are used to align DNA, RNA sequences, and proteins for database searches or structure prediction of protein families [12].

This paper presents a methodology to extract segmented signs from continuous signing. Our approach is based on multiple alignment of videos containing different performances of the same sign. We present a detailed analysis of different features of hand motion and shape, and different alignment algorithms. We use videos recorded from TRT broadcast news for the hearing-impaired. The details of the database are given in Section 2. In Section 3 we explain our particle filter based tracking algorithm to track the face and hands of the signer. Feature extraction techniques used are given in Section 4. The exact start and end locations of the signs are determined via alignment, which is explained in Section 5. Experimental results and conclusions are given in Sections 6 and 7, respectively.

## 2. Database

We use a database of 15 videos 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 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.

For 20 of the words, the corresponding signs are one handed and for the remaining 20, the signs are two handed. Since the presenter is right handed, the one handed signs are performed with the right hand. 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 6, 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 both in tracking, feature extraction and alignment steps.

(a) (b)

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

## 3. Tracking

When signing with natural speed, hands move quite fast and are frequently in contact with each other or with the face. There can also be occlusions where one hand occludes the other or the hand occludes the face. 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 should not be making unrealistic assumptions for the relative locations of the hands and the face. For example in some applications it can be assumed that the left hand will be always on the left of the right hand and vice versa. However this assumption is totally invalid for the case of signing and a tracking algorithm that makes this assumption will definitely fail.

In this work, we use a 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. However, the number of objects may vary: one or both hands may disappear and re-appear. The complexity of the joint PF is reduced by embedding Mean Shift (MS) tracking [11], 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 of the hands and the face and is able to recover fast if the tracking fails. Figure 2 shows a sample frame and the particle distributions with the joint particle filter. We evaluate the performance of our method on a 15 minute signing video. The ground truth for the center of mass coordinates of the hand and face is manually annotated. For each frame, we compared the ground truth with the found positions and achieved a tracking accuracy of 99% for the face and about 96% for the two hands. Details of the tracking algorithm and experiments can be found in [4].

We further analyzed the performance of tracking particularly for the selected samples. The dataset contains a total of 1200 videos for 40 different signs. We manually watched the resulting tracking in these videos and for each video, we decide whether the tracking is acceptable or not. The results of this analysis show that we have an acceptable tracking in 94% of these videos.

# 4. Feature Extraction

## 4.1. Preprocessing

The output of the tracking algorithm is the location, velocity and width, height and orientation of an ellipse that is fitted to each hand and the face. However, to extract accurate hand shape information, we need to find the exact borders of the hands. For this purpose, we apply region growing for each hand, starting from the area obtained from tracking. We limit our region growing with a 50% enlarged ellipse and within this region, we find a mask for the hand. Afterwards, we fit a new ellipse to the resulting hand mask and recalculate the ellipse parameters.

## 4.2. Feature Extraction

We extract six sets of features as illustrated in Figure 3:

- **C,** $\triangle$**C**: Center of mass (CoM) coordinates of the hands and their first order derivatives,
- **E,** $\triangle$**E**: Ellipse parameters for each hand and their derivatives: major and minor axes and rotation angle,
- **D**: Discrete cosine transform coefficients,
- **H**: Histogram of oriented gradients.

## 4.3. Ellipse Parameters

As a simple shape descriptor, we fit an ellipse to the hand images and calculate the ellipse parameters: the center of mass, major and minor axes and the rotation angle. These simple features do not contain any information regarding the hand contour. Instead, they give a rough idea about the shape and orientation of the hand.

We smooth the points along the trajectory of the tracked hand using a moving average filter to eliminate noise. We take the face center of mass as the center of our coordinate system and recalculate the coordinates of the hands to obtain translation invariance. To obtain scale invariance, we normalize the coordinates between 0 and 1. The feature vector sets consist of center positions (**C**) and ellipse parameters (**E**), as well as their first order derivatives ($\triangle$**C**, $\triangle$**E**) for left and right hands.

## 4.4. Discrete Cosine Transform (DCT) coefficients

DCT is used as a feature extractor in many applications such as face [7] and object recognition [3]. DCT expresses the data points in terms of a sum of cosine functions oscillating at different frequencies. When applied on a 2D image, 2D DCT converts the spatial representation in the image into a frequency representation in a 2D matrix. The upper left corner of this resulting matrix contains the low frequency components, which contain most of the information in the image. Thus one can throw away the high frequencies and represent the image with less number of coefficients.

In our method. we use DCT on hand images to extract features for 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 then we scale the image to 20x20 and convert it to gray scale. As most of the information in DCT is concentrated in the lower frequencies, we concentrated on these frequencies. We start from the upper left corner and scan the DCT matrix in a zigzag fashion until we collect 200 coefficients, which is the half of the total coefficients. We eliminate the DC coefficient and reduce the dimensionality with Principal Component Analysis (PCA)

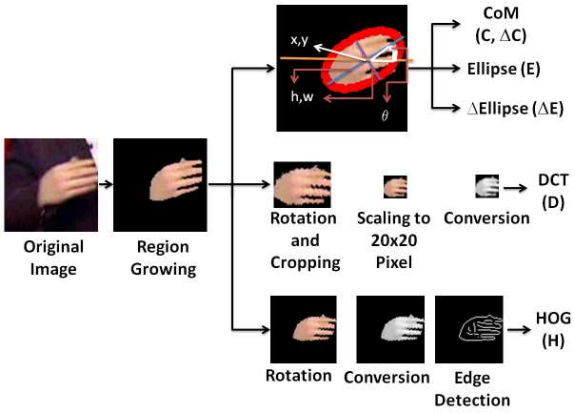



Figure 3: Feature extraction techniques

to obtain vectors with 50 features for each hand. As the dimensionality is high, we use only the features of the right hand and in our experiments we see that omitting the left hand does not effect the system performance significantly.

## 4.5. Histogram of Oriented Gradients (HOG)

HOG descriptors are used in computer vision as feature descriptors in object detection and recognition applications [10], [5]. This method is based on the idea that shape within an image can be described by the distribution of local intensity gradients or edge directions. Hence, it counts occurrences of gradient orientation in localized portions of an image.

In our method, as a preprocessing step, we compensated for the rotation of the hand, translated it to the center of a 80x80 sized box and converted it to gray scale. We find the edges of this image by the Canny edge detector and form gradient histograms. The normalized histogram for the two hands yields an 18-dimensional feature vector.

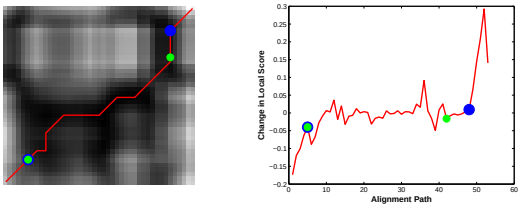
# 5. Alignment

Since speech and signing are not fully synchronized, the speech intervals may not contain the whole sign. To guarantee that the intervals include the sign, we enlarge the intervals from the start and the end, and use the enlarged intervals. When we enlarge our intervals by 15 frames from the start and 5 frames from the end, 96% of the samples are wholly included in the enlarged intervals. Note that, the average duration of the 1200 speech intervals is 15.99 frames. Thus, as a result of the enlargment, we approximately double the length of each interval, in average. Similarly, the average sign duration is 15.72 frames. Consequently, we search for the sign in an interval approximately two times longer than the sign duration.

We used two different alignment techniques: DTW and HMM.

## 5.1. Dynamic Time Warping (DTW)

For each word, we perform the multiple alignment of the sequences via pairwise alignments. The first step in a pairwise alignment is to calculate the local score matrix of the two sequences. We use Euclidean distance as local distance and compute the distance between the feature vectors of each frame from the two sequences. Once the score matrix is calculated we need




(a) (b)

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

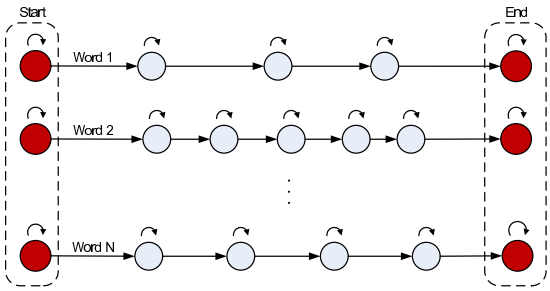



Figure 5: HMM structure used for alignment. A left-to-right HMM is trained for each word where the junk states at the start and end are jointly trained for each model.

to find the alignment path. The general approach in DTW is to calculate the alignment path starting from the upper right or lower left corner. However, in our case we know that there is a high possibility of having junk frames unrelated to the sign at the start and end of the sequences. Starting from the middle increases the possibility of starting from a frame belonging to the sign. Our algorithm first determines a window in the middle of the sequences and selects the position with the lowest score in the window as the starting point. From this point we move forward and backward to find the pairwise alignment path. The start and end locations of the sign are determined by analyzing the changes in the local scores in the alignment path. Local scores decrease when the sign starts and increase when the sign ends. Hence, the start and end locations correspond to the local maxima and local minima of the derivative of the local scores, respectively (see Figure 4). As a result of the pairwise alignments, we obtain 29 candidate start and end locations for each sequence. The final locations are determined by averaging the candidate locations.

### 5.2. Hidden Markov Model (HMM)

HMMs are generative probability models that provide an efficient way of dealing with variable length sequences and missing data [14]. HMMs draw much attention with their 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 have used left-to-right continuous HMMs where observations in each state are modeled via a single Gaussian distribution. The features extracted for the two modalities, hand motion and shape, are combined via feature level fusion, where we concatenate the features of the concurrent modalities in a single feature vector. The training is done with Baum-Welch training with a slight modification in the training of the beginning and end states, which we define as junk states.

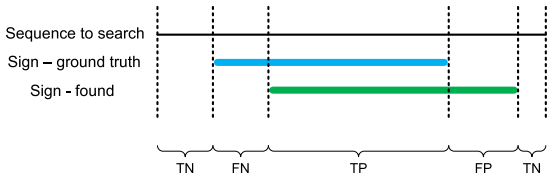



Figure 6: The True Positive (TP), True Negative (TN), False postive (FP) and False Negative (FN) values for the extracted sign with respect to the interval it is searched and the ground truth.

Our aim is to find the part of the sequence that contains only the sign of the word. We model each sequence with a left-to-right HMM model with a state number proportional to the number of frames in the sequence (a ratio of 1/5). We assume that in each sequence, there are junk frames unrelated to the sign at the start and end of the sequence. So, in our model each word has an HMM model containing two junk states, one in the beginning and one in the end. The part that contains the sign is modeled with at least 3 states. If the speech duration is more than 15 frames, the number of states increases proportionally.

For each word in our dataset, we have 30 sequences. The training is performed by leave-one-out cross validation, with one example in the test set and 29 examples in the training set. We train an HMM for each word, such that the start and end states are common for all the sequences of the word (see Figure 5). When the training is over, we find the state sequence for each test sequence using the Viterbi algorithm. 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.

## 6. Experiments

We used three different performance measures to evaluate the system performance: accuracy, precision and recall rates. 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 1 - 3 show the calculation of these measures.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

In Table 1, we analyze the performance of shape features alone and together with motion features on DTW and HMM. We see that ellipse features perform consistently better, either alone or together with the motion features. When we use HMM, we observe a slight increase in accuracy and precision when ellipse features are used. Using $C, E$ as a concatenated feature vector in HMM gives the best performance in general. We see

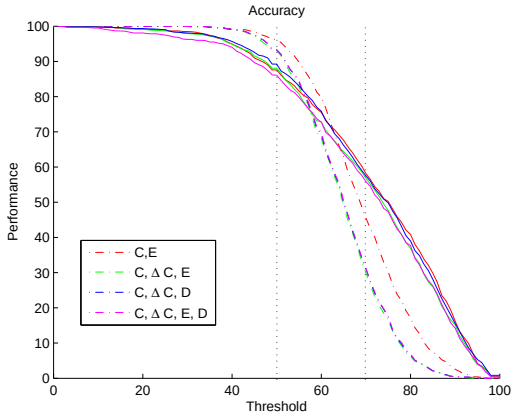



Figure 7: Accuracies of HMM and DTW of different features with respect to different correctness thresholds. Dotted lines are results for DTW, solid lines are results for HMM.

that the effect of the high level shape descriptors, such as DCT and HOG is limited. We think that this is due to the low resolution 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 low resolution and occlusions.

Recall rates are lower for HMM with respect to DTW. This is mainly due to the increase in false negatives in HMM. The segments found by DTW generally cover a wide part of the whole sequence, resulting a high number of true positives, very low number of false negatives but also a high number of false positives. This eventually increases the recall rates. An example alignment can be seen in Figures 8-10, for the words "prime minister", "president", "general". We can see that, even if the true positive values obtained by DTW are higher, false positives are higher as well, with very few false neagtives. Using HMM decreases the errors, especially when the speech and sign synchronisation is poor. In Figure 9, we see that the end points found by DTW are later than the ground truth end points. However, in HMM, end point detection is better with the exception of samples three and five. Further analysis shows that in these examples, the same word, "election", follows the sign "president" and the alignment of HMM can be considered as correct. Consider that we do not use supervised information and the common part in these sequences is "president election".

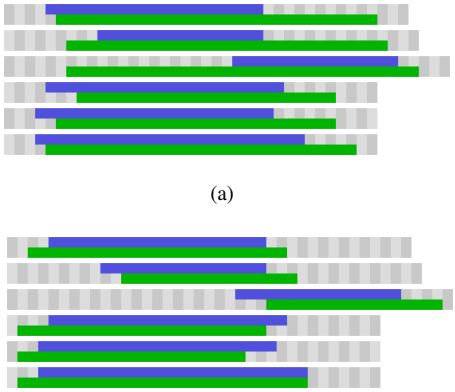
Table 1: Performance of DTW and HMM with respect to accuracy, precision and recall.

| | DTW (%) | | | HMM (%) | | |
|---|---|---|---|---|---|---|
| Feature sets | Acc | Pre | Rec | Acc | Pre | Rec |
| E | 67.50 | 57.85 | 95.35 | 67.82 | 62.27 | 77.61 |
| H | 66.78 | 57.18 | 95.52 | 61.88 | 55.89 | 77.77 |
| D | 67.01 | 57.37 | 95.23 | 56.23 | 53.29 | 48.45 |
| C,E | 68.85 | 59.16 | 94.70 | **71.97** | 66.70 | 83.32 |
| C,H | 67.77 | 58.03 | 95.27 | 61.24 | 54.96 | 75.45 |
| C,D | 67.15 | 57.50 | 95.28 | 53.68 | 47.42 | 49.18 |
| C,△C,E | 64.61 | 55.27 | 97.03 | **70.89** | 65.62 | 83.04 |
| C,△C,H | 64.63 | 55.29 | 96.99 | 67.37 | 61.51 | 82.20 |
| C,△C,D | 64.90 | 55.51 | 96.95 | **71.92** | 69.67 | 75.76 |
| C,△C,E,H | 64.67 | 55.32 | 96.99 | 67.52 | 61.74 | 82.06 |
| C,△C,E,D | 64.99 | 55.58 | 96.96 | **70.15** | 67.76 | 74.38 |

Exact match with the ground truth is rarely possible due

Table 2: Accuracies of DTW and HMM with correctness thresholds 50 and 70

| | th = 50 | | th = 70 | |
|---|---|---|---|---|
| Feature sets | DTW (%) | HMM (%) | DTW (%) | HMM (%) |
| C,E | **96.25** | 87.58 | **45.58** | **58.17** |
| C,△C,E | 93.08 | 88.08 | 29.92 | 56.83 |
| C,△C,D | 93.33 | **89.25** | 31.17 | 57.50 |
| C,△C,E,D | 93.25 | 86.08 | 31.25 | 55.92 |



(a)

(b)

Figure 8: Six examples for the alignment of the word "prime minister" for (a) DTW and (b) HMM, using the feature 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.

to the uncertainty of the sign boundaries. A current study [1] measures performance as follows: When the found signs agree with the ground truth by more than 50%, this is accepted as successful. This corresponds to the signs having the recall rate above 50%. We set the correctness threshold to accuracy rate and investigate the performance under different threshold values. We show the thresholded results on four different feature sets, which have an accuracy of 70% or greater. In Figure 7, we show the behavior of the system when we change the correctness threshold. Table 2 shows the results for correctness thresholds 50 and 70. We observe that for small threshold values DTW outperforms HMM. Yet, when the threshold is larger than 60, the performance of DTW decreases dramatically. When threshold value is 70, the best performance is obtained with HMM using the features C,E with a performance of 58.17%. This means that, for 58.17% of the alignments obtained with HMM, the ratio between the error and the searched window is less than 30%.

## 7. Conclusions

Automatic extraction of sign databases or sign dictionaries is a newly arising area of research given the newly available sources of information. Broadcast news for the hearing-impaired provides an excellent data source as the signs and the speech are performed in parallel and the locations that are found by the speech indicate approximate locations for the signs. This enables the extraction of isolated signs from continuous sign-

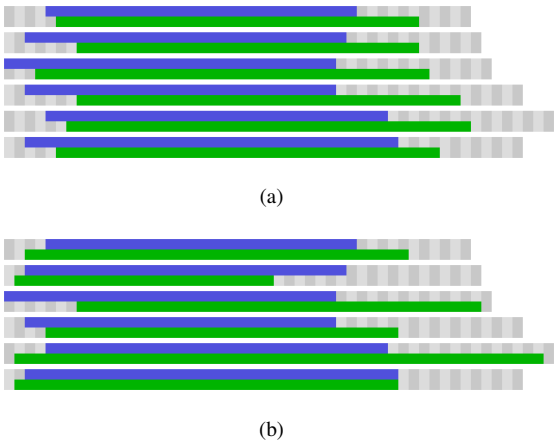


Figure 9: Six examples for the alignment of the word "president" for (a) DTW and (b) HMM, using the feature C,E. Each box stands for one frame. The sign is searched within the gray area, the green lines represent the found segment, and the blue lines represent the ground truth.

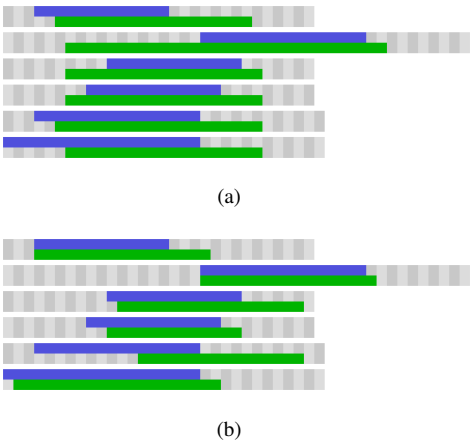


Figure 10: Six examples for the alignment of the word "general" for (a) DTW and (b) HMM, using the feature C,E. Each box stands for one frame. The sign is searched within the gray area, the green lines represent the found segment, and the blue lines represent the ground truth.

ing without the need for pre-learned models of the signs, allowing to automatically produce automatic sign dictionaries or databases.

Our approach is based on multiple alignment of sequences, which all contain the same sign that we search for, but different performances. Although these sequences contain the sign, these are broad intervals and we need to find the exact location of the signs. We aim to find the longest common subsequence in these multiple sequences, which gives us the sign. Our experiments show that modeling the hand motion and shape using HMMs gives the best result.

## 8. Acknowledgement

This work is supported by Tübitak project 107E021 and Tübitak - RFBR joint project 108E113.

## 9. References

[1] J. Alon, Q. Yuan, and S. Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 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 *Proceedings of International Conference on Graphics, Vision and Image Processing (GVIP-05*, pages 362–368, 2005.

[4] 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 (eNTERFACE08), Paris, France*, pages 24–33, 2008.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition, USA*, volume 2, pages 886–893, 2005.

[6] T. Darrell, I. A. Essa, and A. P. Pentland. Task-specific gesture analysis in real-time using interpolated views. *Pattern Analysis and Machine Intelligence*, 18(12):1236–1242, 1996.

[7] H. K. Ekenel and R. Stiefelhagen. Local appearance based face recognition using discrete cosine transform. In *13th European Signal Processing Conf., Antalya, Turkey*, 2005.

[8] G. Fang, W. Geo, and D. Zhao. Large-vocabulary continuous sign language recognition based on transition-movement models. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans*, 37(1):1–9, 2007.

[9] A. Farhadi and D. Forsyth. Aligning asl for statistical translation using a discriminative word model. In *Computer Vision and Pattern Recognition*, volume 2, pages 1471–1476, 2006.

[10] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. In *In International Workshop on Automatic Face and Gesture Recognition*, pages 296–301, 1995.

[11] E. Maggio and A. Cavallaro. Hybrid particle filter and mean shift tracker with adaptive transition model. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), USA*, 2005.

[12] C. Notredame. Recent progresses in multiple sequence alignment: a survey. *Pharmacogenomics*, 3(1):131–144, January 2002.

[13] S. Parlak and M. Saraclar. Spoken term detection for Turkish broadcast news. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), USA*, 2008.

[14] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77 of *no. 2*, pages 257–285, 1989.

[15] U. Zeshan. Aspects of Türk isaret dili (Turkish sign language). *Sign Language & Linguistics*, 6(1):43–75, 2003.