SignTutor: An Interactive System for Sign Language Tutoring

Oya Aran[1], Ismail Ari[1], Alexandre Benoit[2], Pavel Campr[3], Ana Huerta Carrillo[4], François-Xavier Fanard[5], Lale Akarun[1], Alice Caplier[2], Michele Rombaut[2] and Bulent Sankur[1]

[1]*Bogazici University,* [2]*LIS_INPG,* [3]*University of West Bohemia in Pilsen,* [4]*Technical University of Madrid,* [5]*Universite Catholique de Louvain*

***Abstract*—**Sign Language, the natural communication medium for a deaf person, is difficult to learn for the general population. The prospective signer should learn specific hand gestures in coordination with head motion, facial expression and body posture. Since language learning can only advance with continuous practice and corrective feedback, we have developed an interactive system, called SignTutor, which automatically evaluates users' signing and gives multimodal feedbacks to guide them to improve their signing. SignTutor allows users to practice instructed signs and to receive feedback on their performance. The system automatically evaluates sign instances by multimodal analysis of the hand and head gestures. The time and gestural variations among different articulations of the signs are mitigated by the use of hidden Markov models. The multimodal user feedback consists of a text-based information on the sign, and a synthesized version of the sign on an avatar as a visual feedback.. We have observed that the system has a very satisfactory performance, especially in the signer-dependent mode, and that the user experience is very positive.

***Index Terms*—Hand and head gestures, sign language recognition, HMM fusion, user feedback**

I. Introduction

Sign language recognition (SLR) is a multidisciplinary research area involving pattern recognition, computer vision, natural language processing and linguistics. It is a multifaceted problem not only because of the complexity of the visual analysis of hand gestures but also due to the highly multimodal nature of sign languages. Although sign languages are well-structured languages with a phonology, morphology, syntax and grammar, they are different from spoken languages: The structure of a spoken language makes use of words sequentially, whereas a sign language makes use of several body movements in parallel. The linguistic characteristics of sign language are different from those of spoken languages due to the existence of several components affecting the context, such as the use of facial expressions and head movements in addition to the hand movements [1][2].

A. Automatic Sign language Recognition

A brief survey of sign language grammars illustrates the challenges faced: Sign language phonology makes use of the hand shape, place of articulation, and movement. The morphology uses directionality, aspect, and numeral incorporation, and syntax uses spatial localization and agreement as well as facial expressions. The whole message is contained not only in hand gestures and shapes (manual signs) but also in facial expressions and head/shoulder motion (non-manual signs). As a consequence, the language is intrinsically multimodal. In sum, sign language recognition is a very complex task: a task that uses hand shape recognition, gesture recognition, face and body parts detection, facial expression recognition as basic building blocks [3].

Frontier research on hand gesture recognition and on sign language recognition (SLR) has mainly used instrumented gloves, which provide accurate data for hand position and finger configuration. These systems require users to wear cumbersome devices on their hands. However, humans would prefer systems that operate in their natural environment. Since the mid 90's, improvements in camera hardware have enabled real-time vision-based hand gesture recognition [4]. Instead of using instrumented gloves, vision based-systems, which only require one or more cameras connected to the computer, have been adopted. While vision-based SLR systems provide a

more user-friendly environment, they also introduce several new challenges, such as the detection and segmentation of the hand and finger configuration, or handling occlusions.

Signs comprise dynamical elements. A recognition system that focuses only on the static aspects of the signs has a limited vocabulary. Hence, for recognizing hand gestures and signs, one must use methods that are capable of modeling inherent temporal characteristics of the gestures. Researchers have used several methods such as neural networks, hidden Markov models (HMM), Dynamic Bayesian Networks (DBN), Finite State Machines (FSM) or template matching [5]. Among these methods, HMMs have been used the most extensively and have proven successful in several kinds of SLR systems. Initial studies on vision-based SLR focused on limited vocabulary systems, which could recognize 40-50 signs. These systems were capable of recognizing isolated signs and also continuous sentences, but with constrained sentence structure [6]. The scalability problem is addressed in the following studies, where an approach based on identifying phonemes/components of the signs [7] rather than the whole sign has been adopted. The advantage of identifying components is the decrease in the number of subunits that should be trained, which in turn will be used to constitute all the signs in the vocabulary. With component-based systems, large vocabulary SLR can be achieved to a certain degree.

Another important aspect of sign language is all the non-manual components concomitant with hand gestures. To give an idea, in an instrumented glove-based system, a database of 5119 signs has been recognized [8]. Most of these SLR systems concentrate on hand gesture analysis only. However, without integrating non-manual signs, it is not possible to interpret the meaning of all these signs. There are only a limited number of studies that integrate non-manual and manual cues for SLR [3]. Current multimodal SLR systems either integrate lip motion and hand gestures, or only classify and incorporate either the facial expression [9] or the head motion [10].

Recognizing unconstrained continuous sign sentences is another challenging problem in SLR. The significance of co-articulation effects necessitates the usage of language models similar to the ones that are used in speech recognition. These are complex problems to be tackled and require adequate know-how and technology such as high-speed with higher-resolution than the commercial cameras. An ideal automatic SLR system should be able to accurately recognize a large vocabulary set enacted in continuous sentences. Such a system should operate in real time, and must be robust to lighting, illumination and other environmental conditions. It should exploit both the manual and non-manual signs and the grammar and syntax of the sign language.

There are many potential application areas of SLR systems that require sign-to-text translation. These applications include HCI (Human-Computer Interaction), public information access such as kiosks, or translation and dialog systems for human-to-human communication. An interesting application area, coupled with sign synthesis capability, is in communicating sign data over a channel. On the sender side, the sign is captured and the SLR code is sent to the receiving end where it is synthesized and displayed on an avatar. This scheme would be more flexible and more bandwidth efficient as compared to sending the captured sign videos.

B. SLR assisted Sign Language Education

Practice can significantly enhance the learning of a language when there is validation and feedback. This is true for both spoken and sign languages. For spoken languages, students can evaluate their own pronunciation and improve to some extent by listening to themselves. Similarly, sign language teachers suggest that their students practice in front of the mirror. Instead with an SLR-based system, students can practice by themselves, validate and evaluate their signing. Such a system would be called such as SignTutor, which would be instrumental in assisting sign language education, especially for non-native signers.

SignTutor aims to teach the basics of the sign language interactively. The advantage of a SignTutor is that it automatically evaluates the student's signing and enables auto-evaluation via visual feedback and information about the goodness of the performed sign. The interactive platform of SignTutor enables the users to watch and learn new signs, to practice and to validate their performance. The SignTutor automatically evaluates the users' signing and communicates them the outcome in various feedback modalities: a text message, the recorded video of the user, the video of the segmented hands and/or an animation on an avatar.

One of the key factors of SignTutor is that it integrates hand motion and shape analysis together with head

motion analysis to recognize signs that include both hand gestures and head movements. This is very important since head movements are one aspect of sign language that most students find hard to perform in synchronism with hand gestures. To put this advantage of SignTutor, we have dwelled mostly on the signs that have similar hand gestures and that are mainly differentiated by head movements. In view of the prospective users and the usage environment of SignTutor, we have opted for a vision-based user-friendly system which can work with easy to obtain equipment, such as webcams. The system operates with a single camera focused to the upper body of the user.

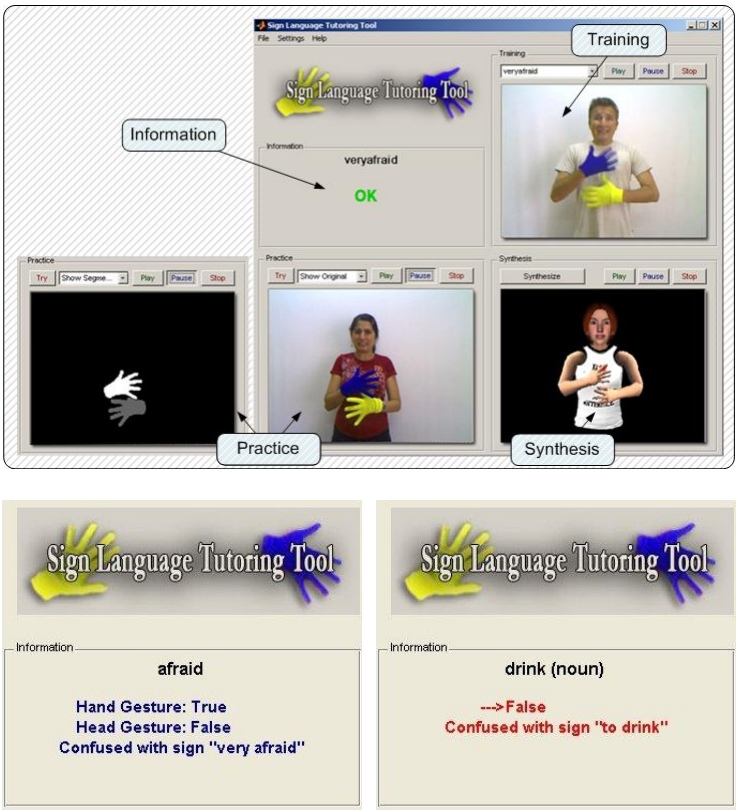


Figure 1. SignTutor GUI: training, practice, information, synthesis panels and feedback examples

Figure 1 shows the graphical user interface of Sign Tutor. The system follows three steps for teaching a new sign: Training, practice and feedback. For the training phase, SignTutor provides a pre-recorded reference video for each sign. The users select a sign from the list of possible signs and watch the pre-recorded instruction video of that sign until they are ready to practice. In the practice phase, users are asked to perform the selected sign and their performance is recorded by the webcam. For an accurate analysis, users are expected to face the camera, with full upper body in the camera field of view. The recording continues until both hands are out of the camera. SignTutor analyzes the hand motion, hand shape and head motion in the recorded video and compares it with the selected sign.

We have integrated several modalities to the system for giving feedback to the user as for the quality of the enacted sign. The goodness criteria are given separately for the two components: the manual component (hand motion, shape, and position) and the non-manual component, (head and facial feature motion), together with the sign name with which it is confused (see Figure 1). Users can watch the video of their original performance. If the

sign is properly performed, users may watch a caricaturized version of their performance on an animated avatar. A demonstration video of SignTutor can be downloaded from [11].

In summary, SignTutor aims to facilitate sign language learning especially for non-native beginners, by providing an interactive system. To assess the usability of the overall system, we have performed a user study, with the students of the Turkish Sign Language beginner level course given in Boğaziçi University. We have collected the test scores and the comments of users to gauge its effectiveness.

The rest of the paper evolves as follows. SignTutor modules are described in Section 2. In Section 3, the results of experiments conducted to assess the performance of the system are presented. Section 4 presents the user study and discusses its results. We conclude and discuss future work in Section 5.

II. SIGNTUTOR MODULES

The block diagram of the SignTutor consists a face and hand detector stage, followed by the analysis stage, and the final sign classification box as illustrated in Figure 2. The critical part of SignTutor is the analysis and recognition sub-system which receives the camera input, detects and tracks the hand, extracts features and classifies the sign. In the sequel, we present the analysis and recognition modules and describe the synthesis and animation subsystem, which aims to provide a simple visual feedback environment for the users.

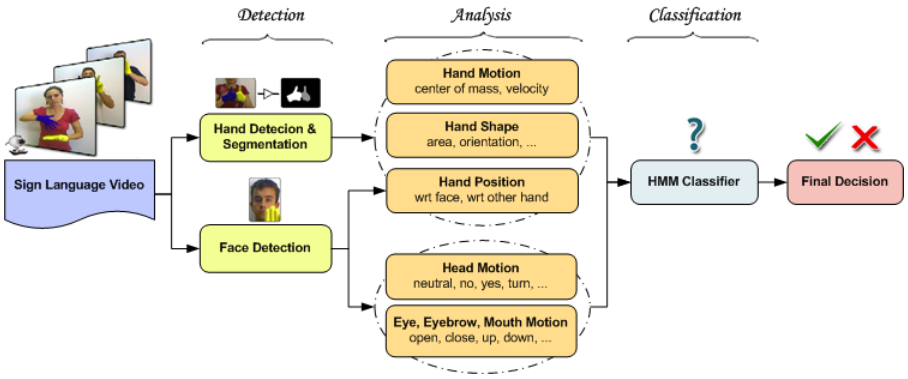


Figure 2. SignTutor system flow. Detection, analysis and classification steps.

A. Hand detection and segmentation

Although skin color features can be applied for hand segmentation in controlled illumination, segmentation becomes problematic when skin regions overlap and occlude each other. In fact in sign language, hand positions are often near the face and sometimes have to be in front of the face. Hand detection, segmentation and occlusion problems are simplified when users wear colored gloves. The use of a simple marker as a colored glove makes the system robust to changing background and illumination conditions.

For each glove color, we train its respective histogram of color components using several images. We have chosen HSV color space due its robustness to changing illumination conditions [12]. The H, S and V components are divided into bins. At each bin of the histogram, we calculate the number of occurrences of pixels that correspond to that bin, and finally the histogram is normalized. To detect hand pixels in a scene, we find the histogram bin it belongs to and apply thresholding. We apply double thresholding, set at low and high values, to ensure connectivity: A pixel is considered as a hand pixel if its histogram value is higher than the high threshold, or if it is in between the two thresholds it is still labeled as glove (hand) provided one or more of neighbor pixels were labeled as glove. The final hand region is assumed to be the largest connected component over the detected pixels.

B. Analysis of hand motion

The system processes the motion of each hand by tracking its center of mass (CoM) and estimating in every frame the position and velocity of each segmented hand. However, the hand trajectories can be corrupted by segmentation noise. Moreover, hands may occlude each other or there may be sudden changes in lighting (e.g., a room light turned on or off), which may result in detection and segmentation failures. Thus, we use two independent Kalman filters, one for each hand, to smooth the estimated trajectories. The motion of each hand is approximated by a constant velocity motion model, hence acceleration is neglected. When the system detects a hand in the video, it initializes the corresponding Kalman filter. Before each sequential frame, Kalman filter predicts the new hand position, and the filter parameters are updated with the hand position measurements found by the hand segmentation step. We calculate the hand motion features from the posterior states of the corresponding Kalman filter: x, y coordinates of CoM and velocity [13]. When there is a detection failure due to occlusion or bad lighting, we only use the Kalman filter prediction without updating the filter parameters. Finally, the system assumes that the hand is out of the camera view if no hand segment can be detected for a number of consecutive frames.

The trajectories must be further normalized to obtain translation and scale invariance. We use a normalization strategy similar to [13]. The normalized trajectory coordinates are calculated via min-max normalization. The translation normalization is handled by calculation the mid points of the range of x and y coordinates, denoted as x_m, y_m. The scaling factor, d, is selected to be the maximum of the spread in x and y coordinates, since scaling horizontal and vertical displacements with different factors disturbs the shape. The normalized trajectory coordinates, $(<x'_1;y'_1>;...;<x'_t;y'_t>; ...;<x'_N;y'_N>)$ such that $0 <= x'_t , y'_t <= 1$, are then calculated as follows:

$$x'_t = 0.5 + 0.5\ (x_t - x_m) / d$$
$$y'_t = 0.5 + 0.5\ (y_t - y_m) / d$$

Since signs can also be two handed, both hand trajectories must be normalized. However, normalizing the trajectory of the two hands independently may result in a possible loss of data. To solve this problem, the midpoint and the scaling factor of the left and right hand trajectories are calculated jointly. Following the trajectory normalization, the left and right hand trajectories are translated such that their starting position is at (0,0).

C. Extracting features from a 2D hand shape

Hand shape and finger configuration contribute significantly to sign identification, since each sign has a specific movement of the head, hands and hand postures. Moreover, there are signs which solely depend on the hand shape. Our system is intended to work with a single low-resolution camera whose field of view covers the upper body of the user, hence is not directly focused on the hands. In this setup, we face several difficulties:

- Low resolution of hand images, where each hand image is smaller than 80x80 pixels,
- Segmentation errors due to blurring caused by fast movement
- More than one hand posture can result in the same binary image, that is, silhouette.

These problems constrain us to use only low-level features, which are robust to segmentation errors and work well with low resolution images. Therefore we use simple appearance-based shape features calculated from the hand silhouettes. The features are selected to reflect differences in hand shape and finger postures. They are also required to be scale invariant so that hands with similar shape but different size result in the same feature values. However recall that our system uses a single camera, hence we do not have depth information, except for the foreshortening due to perspective. In order to maintain some information about the z-coordinate (depth), five of the 19 features were not scale normalized. Prior to the calculation of the hand shape features, we take the mirror reflection of the right hand so that we analyze both hands in the same geometry; with thumb to the right. All 19 features are listed in TABLE I. It is important to note that these features are not invariant to viewpoint, and the users are required to sign facing the camera for an accurate analysis. The classifier is tolerant of small rotations that can naturally occur while signing.

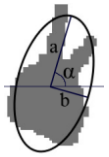
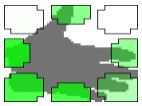
Seven of the features (#1,2,4,5,6,7,8) are based on using the best fitting ellipse (in the least-squares sense) to the

hand silhouette. The inclination angle assumes values in the range [0, 360]. In order to represent the 4-fold symmetry of the ellipse, we use *sin(2*α)* and *cos(2*α)* as features, where α is in the range [0, 180].

Features #9 to 16 are based on using "area filters". The bounding box of the hand is divided into eight areas, in which, percentage of hand pixels are calculated. Other features in TABLE I are perimeter, area and bounding box width and height.

All 19 hand shape features are normalized into values between 0 and 1. This is obtained by dividing the percentage features (#9 to 16) by 100, and the cardinal number features by their range, that is, by using $F_n = (F - min) / (max - min)$ where *min* is minimum value of the feature in the training dataset and *max* is maximum value. Any value exceeding the [0,1] interval is truncated.

TABLE I. HAND SHAPE FEATURES

#		Feature	Invariant	
			Scale	Rotation
1	 *[best fitting ellipse]*	Best fitting ellipse width		✓
2		Best fitting ellipse height		✓
3		Compactness (perimeter^2/area)	✓	✓
4		Ratio of hand pixels outside / inside of ellipse	✓	✓
5		Ratio of hand / background pixels inside of ellipse	✓	✓
6		sin (2*α) α = angle of ellipse major axis	✓	
7		cos (2*α) α = angle of ellipse major axis	✓	
8		Elongation (ratio of ellipse major/minor axis length)	✓	✓
9	 *[area filters] white: areas without hand green: areas with hand*	Percentage of NW (north-west) area filled by hand	✓	
10		Percentage of N area filled by hand	✓	
11		Percentage of NE area filled by hand	✓	
12		Percentage of E area filled by hand	✓	
13		Percentage of SE area filled by hand	✓	
14		Percentage of S area filled by hand	✓	
15		Percentage of SW area filled by hand	✓	
16		Percentage of W area filled by hand	✓	
17		Total area (pixels)		✓
18		Bounding box width		
19		Bounding box height		

D. Analysis of head movements

Once the face is detected [14] rigid head motions such as head rotations and head nods are determined by using an algorithm inspired by the human visual system. First, we apply a filter following the model of the human retina [15]. This filter enhances moving contours with Outer Plexiform Layer (OPL) and cancels static ones with Inner Plexiform Layer (IPL). This prefiltering mitigates any illumination changes and noise. Second, we compute the fast Fourier transform (FFT) of the filtered image in the log polar domain as a model of the primary visual cortex (V1) [16]. This step allows extracting two types of features: the quantity of motion and motion event alerts. In parallel, an optic flow algorithm extracts both orientation and velocity information only on the motion event alerts issued by the visual cortex stage [17]. Thus after each motion alert, we estimate the head velocity at each frame. Figure 3 gives the block diagram of the algorithm. After these three steps, the head analyzer is outputs three features per frame: the quantity of motion, and the vertical and horizontal velocities.

These three features provided by the head motion analyzer can vary with different performances of the same sign. Moreover, head motion is not directly synchronized with the hand motion. To handle the inter- and intra-subject differences, weighted average smoothing is applied to head motion features, with $\alpha = 0.5$. The smoothed head motion feature vector at time i, F_i, is calculated as $F_i = \alpha F_i + (1 - \alpha) F_{i-1}$. This smoothing has the effect of mitigating the noise between different performances of a sign and creating a slightly smoother pattern.

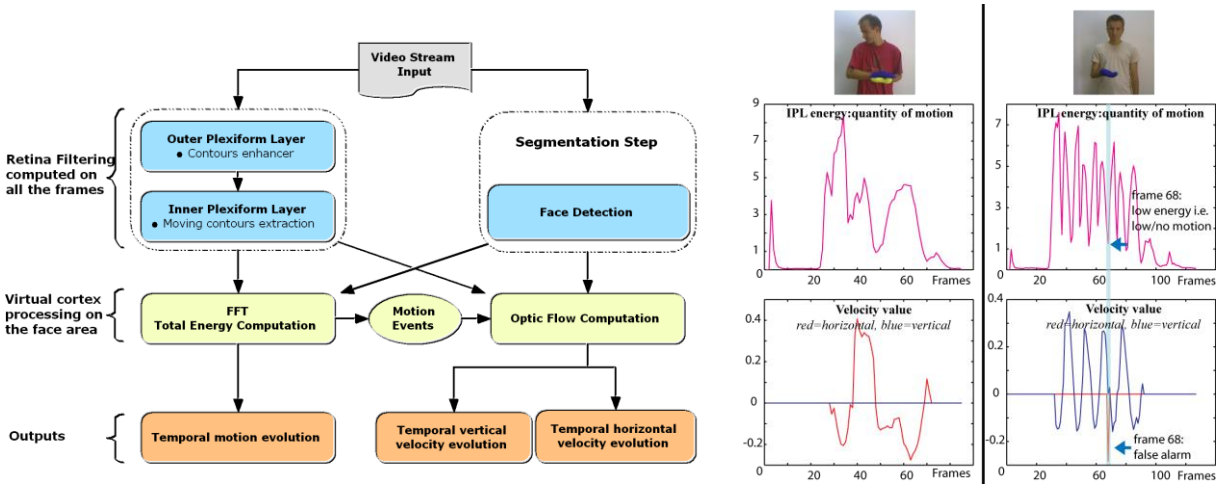


Figure 3. Algorithm for rigid head motion data extraction

E. Preprocessing of sign sequences

The video sequences obtained contain frames where the signer is not performing any sign (beginning and terminating parts) and some frames that can be considered as transition frames. These frames of the sequence are eliminated by considering the result of the hand segmentation step:

- All frames at the beginning of the sequence are eliminated until a hand is detected.
- If the hand fails to be detected during the sequence for less than N consequent frames, the shape information is copied from the last frame where there was still detection to the current frame.
- If the hand fails to be detected for more than N consequent frames, the sign is assumed to be finished. The remaining frames including the last N frames are eliminated.
- After these prunings, transition frames, defined as the T frames from the start and end of the sequence, are deleted.

F. Cluster based classification: A sequential fusion approach

At the classification phase, we have used HMMs to model each sign and the classification decision is given via the likelihood of the observation sequence with respect to each HMM. HMMs are used in many areas such as speech recognition and bioinformatics, for their capability of modeling variable length sequences and dealing with temporal variability within similar sequences [18]. HMMs are therefore preferred in gesture and sign recognition as the changes in the speed of the performed sign or slight changes in the spatial domain are successfully handled.

We use a sequential fusion method for combining manual and non-manual parts of the sign. The strategy uses the fact that there may be similar signs which differ slightly and cannot be classified accurately in an "all signs" classifier. These signs form a *cluster* and their intra-cluster classification must be handled specially. Thus, our sequential fusion method is based on two successive classification steps: In the first step, we perform an inter-cluster classification and in the second step we do intra-cluster classifications. Since we want our system to be as

general as possible and open-ended to new signs without re-designing it, we do not use any prior knowledge about the sign clusters. We let the system discover potential sign clusters, that are similar in manual gestures, but that differ in non-manual motions. Instead of rule-based programming of these signs, we opt to extract the cluster information as a part of the recognition system, as described below.

We give the base decision by a general model, $HMM_{M\&N}$, which uses both hand and head features in the same feature vector. However, this approach suffers from the curse of dimensionality and the head information is not utilized well. We utilize the head information in a dedicated model, HMM_N, and used the likelihoods of this model to give the final decision.

The system uses the trained HMM models. The training is two-fold:

- Train the HMM models for each sign, models for both $HMM_{M\&N}$ and $HMM_{N.}$
- Extract the cluster information via the joint confusion matrix of $HMM_{M\&N}$. This joint confusion matrix is formed by summing the confusion matrices of the validation sets in a cross validation stage. We investigate the misclassifications by using the joint confusion matrix. If all samples of a sign class are correctly classified, the cluster of that sign class only contains itself. Otherwise for each misclassification, we mark that sign as potentially belonging to the cluster. The marked signs become members of the cluster when the number of misclassifications exceeds a given threshold. We use 10% of the total number of validation examples as the threshold value.

The fusion strategy for a test sample is as follows:

- Calculate the likelihoods of $HMM_{M\&N}$ for each sign class and select the sign class with the maximum likelihood as the base decision.
- Send the selected sign and its cluster information to HMM_N.
- Combine the likelihoods of $HMM_{M\&N}$ and HMM_N with the sum rule and select the sign with the maximum likelihood as the final decision.

The classification module receives all the features calculated in the hand and head analysis modules as input. For each hand, there are four hand motion features (position and velocity in vertical and horizontal coordinates), 19 hand shape features and three head motion features per frame. We also use the relative position of the hands with respect to the face CoM, yielding two additional features (distance in vertical and horizontal coordinates) per hand per frame. We normalize these latter distances by the face height and width, respectively, and use the normalized and pre-processed sequences to train the HMM models. Each HMM model is a continuous 4-state left-to-right model and is trained for each sign, using the Baum-Welch algorithm.

G. Visual feedback via a synthesized avatar

As one of the feedback modalities, SignTutor provides a simple animated avatar that mimics the users' performance for the selected sign. The synthesis is based on the features extracted in the analysis part. Currently the avatar only mimics the head motion of the user and hand motions are animated from a library.

Our head synthesis system is based on the MPEG-4 Facial Animation Standard [19]. As input, it receives the motion energy, the vertical and horizontal velocities of the head motion and the target sign as well. It then filters and normalizes these data in order to compute the head motion during the considered sequence. The result of the processing is expressed in terms of Facial Action Points (FAP) and is fed into the animation player. For head animation, we use XFace [20], an MPEG-4 compliant 3D talking head animation player. The hands and arms synthesis system is based on OpenGL, and an animation is prepared explicitly for each sign. We merge the head animation with the hands and arms animation to form an avatar with full upper body.

III. EVALUATION OF THE SYSTEM ACCURACY

We evaluate the recognition performance of our system on a dataset of 19 signs from ASL. These signs are selected to emphasize the usage of non-manual signs, the effect of the small modifications of manual signs and

hand-head coordination. There are eight base signs that represent words and their systematic variations in the form of non-manual signs, or inflections in the signing of the same manual sign [3]. Table II lists the signs used for evaluation. For each sign, we have recorded five repetitions from eight subjects. In Table II, base signs and their variants are shown with the type of variation, either a hand motion variation or a non manual sign or both. For the rest of this paper, we assume that the base sign and its variants form a semantic cluster and we will call these clusters as *base sign clusters*.

TABLE II: ASL SIGNS USED IN SIGNTUTOR

Base Sign	Variant	Variation on hand motion	Head Motion
Clean	Clean		
	Very clean		✓
afraid	Afraid		
	Very afraid	✓	✓
Fast	Fast		
	Very fast		✓
drink	To drink		✓
	Drink (noun)	✓	
open (door)	To open		
	door (noun)	✓	
Here	[smbdy] is here		✓
	Is [smbdy] here?		✓
	[smbdy] is not here		✓
Study	Study		
	Study continuously	✓	✓
	Study regularly	✓	✓
Look at	Look at		
	Look at continuously	✓	✓
	Look at regularly	✓	✓

We compare three different classifiers to show the effect of using the non-manual information with different fusion techniques on the classification accuracy.

1) Classification by using only manual information

This classification is done via HMMs that are trained only with the hand gesture information related to the signs. Since hands form the basis of the signs, these models are expected to be very powerful in classification. However, absence of the head motion information precludes correct classification when signs differ only in head motion. We denote these models as HMM_M.

2) Feature fusion of manual and non-manual information

The manual information and the non-manual information can be combined in a single feature vector to jointly model the head and hand information. Since there is not a direct synchronization between hand and head motions, these models are not expected to have much better performance than HMM_M. However using head information results in a slight increase in the performance. We indicate these models as $HMM_{M\&N}$.

3) Sequential fusion of manual and non-manual information

The aim of this fusion approach, as explained in the previous section, is to apply a two-tier cluster-based sequential fusion strategy. The first step identifies the cluster of the performed sign within a general model, $HMM_{M\&N}$, and the confusion inside the cluster is resolved at the second step, with a dedicated model, HMM_N, which uses only head information. The head motion is complementary to the sign thus it cannot be used alone to classify the signs. However, in the sequential fusion methodology, indicated as $HMM_{M\&N} \rightarrow HMM_N$, they are used to perform intra-cluster classification. We report the results of our sequential fusion approach with different clusters, first on base sign clusters and then on automatically identified clusters based on the joint confusion matrix.

The merit of a recognition system is its generalization power of a learned concept to new instances. In sign language recognition, there are two kinds of new instances for any given sign: (1) signing of a signer whose other signing videos are put in the training set; (2) signing of a new signer. The former is called as the signer dependent and the latter as the signer independent experiments. The real performance of a sign language recognition system must be measured in a signer-independent experiment.

We report two sets of results: the base sign accuracy and the overall accuracy. To report the base sign accuracy, we assumed that a classification decision is correct if the classified sign and the correct sign are in the same base sign cluster. The base sign accuracy is important for the success of our sequential fusion method based on sign clusters. The overall accuracy reports the actual accuracy of the classifier over all the signs in the dataset. These accuracies are reported on the two protocols: the signer-independent protocol and the signer-dependent protocol.

B. Signer-Independent Protocol

In the signer-independent protocol, we constitute the test sets from instances of a group of subjects in the dataset and train the system with sample signs from the rest of the signers. For this purpose, we apply an 8-fold cross-validation, where at each fold test set consists of instances from one of the signers and the training set consists of instances from the other signers. In each fold there are 665 training instances and 95 test instances. The results of signer-independent protocol are given in Table III.

TABLE III. SIGNER-INDEPENDENT TEST RESULTS

	Sbj #1	Sbj #2	Sbj #3	Sbj #4	Sbj #5	Sbj #6	Sbj #7	Sbj #8	Average
					Base Sign Accuracy				
HMM_M	100.0	100.0	100.0	100.0	96.84	100.0	100.0	100.0	99.61
$HMM_{M\&N}$	100.0	100.0	100.0	100.0	97.89	100.0	100.0	100.0	99.74
$HMM_{M\&N} \rightarrow HMM_N$	100.0	100.0	100.0	100.0	97.89	100.0	100.0	100.0	99.74
					Overall Accuracy				
HMM_M	66.32	77.90	60.00	71.58	57.90	81.05	52.63	70.53	67.24
$HMM_{M\&N}$	73.68	91.58	71.58	81.05	62.11	81.05	65.26	77.89	75.53
$HMM_{M\&N} \rightarrow HMM_N$ *Base Sign Clusters*	82.11	72.63	73.68	88.42	56.84	80.00	81.05	71.58	75.79
$HMM_{M\&N} \rightarrow HMM_N$ *Automatically Identified Clusters*	85.26	76.84	77.89	89.47	63.16	80.00	88.42	75.79	**79.61**

The base sign accuracies of each of the three classifiers in each fold are 100% except for the fifth signer, which is slightly lower. This performance result shows us that a sequential classification strategy is appropriate, where specialized models are used in a second step to handle any intra-cluster classification problem.

The need for the usage of the head features can be deduced from the high increase of the overall accuracy with the contributions of non-manual features. With $HMM_{M\&N}$, the accuracy increases to 75.5% as compared to the accuracy of HMM_M, 67.2%. Further increase is obtained by using our sequential fusion methodology with automatically defined clusters. We also report the accuracy of the sequential fusion with the base sign clusters, to show that using those semantic clusters results in a 4% lower accuracy than automatic clustering.

C. Signer-Dependent Protocol

In the signer-dependent protocol, we put examples from each subject in both of the test and training sets, although they never share the same sign instantiation. For this purpose, we apply a 5-fold cross validation where at each fold and for each sign, four out of the five repetitions of each subject are placed into the training set and the remaining one to the test set. In each fold there are 608 training examples and 152 test examples. The results of signer dependent protocol are given in Table IV.

The base sign accuracies of each of the three classifiers are similar to the signer-independent results. The overall accuracies become much higher and the sequential fusion technique does not anymore contribute significantly. This is probably a result of the ceiling effect and the differences between the approaches are not apparent as a result of the already high accuracies.

TABLE IV. SIGNER-DEPENDENT TEST RESULTS

	Fold #1	Fold #2	Fold #3	Fold #4	Fold #5	Average
			Base Sign Accuracy			
HMM_M	99.34	100.0	98.68	100.0	100.0	99.61
$HMM_{M\&N}$	100.0	100.0	98.68	100.0	100.0	99.74
$HMM_{M\&N} \rightarrow HMM_N$	100.0	100.0	98.68	100.0	100.0	99.74
			Overall Accuracy			
HMM_M	92.76	89.47	89.47	94.08	92.76	91.71
$HMM_{M\&N}$	92.76	95.39	93.42	96.05	94.08	94.34
$HMM_{M\&N} \rightarrow HMM_N$ *Base Sign Clusters*	92.76	95.39	92.76	95.39	94.08	94.08
$HMM_{M\&N} \rightarrow HMM_N$ *Automatically Identified Clusters*	92.76	95.39	92.76	96.05	94.08	94.21

IV. USER STUDY

We have conducted a user study to measure the real-life performance of SignTutor and to assess the usability of the overall system. Our subjects were volunteers, two males and four females, six out of seven students taking an introductory Turkish Sign Language course given in Bogazici University. Two of the students (one male and one female) are from Computer Engineering department and the rest are from the Foreign Language Education department. All subjects were highly motivated for the experiment and were excited about the SignTutor when they were first told about the properties of the system.

We performed the experiments in two sessions, where in each session, subjects were asked to practice three signs. The second session was conducted with a time lapse of at least one week after the first session. Before the first session, we present the SignTutor interface to the subjects in the form of a small demonstration. At the second session, the users are expected to start using the SignTutor without any presentation. For each subject, we measured the time on task, where each task is defined as the learning, practicing and evaluating one of the three signs. At the end of the experiment, we interviewed the subjects and asked them to fill a small questionnaire.

We asked three questions in the questionnaire and the subjects scored at five levels, from strongly disagree (1) to strongly agree (5). The results are shown in Figure 4. The average scores for all questions are above four, indicating the favorable views of the subjects on the usability of the system.

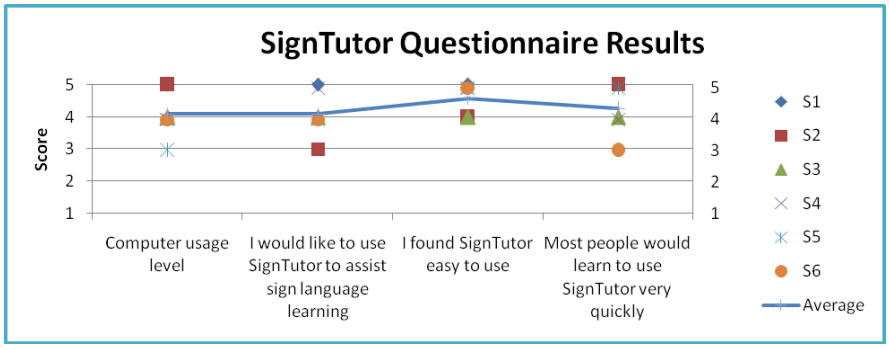


Figure 4. Usability questionnaire results

In session 1, the subjects are asked to practice three signs: AFRAID, VERY AFRAID and DRINK (NOUN). These three signs are selected such that the first two are variants of the same base sign, performed by two hands; and the third is a completely different sign, performed by a single hand. We measured the number of seconds for each task and the results are shown in Figure 5. The subjects practiced each sign until they receive a positive feedback. The average number of trials is around two. The average time is 85 seconds for the first task and decreases to 60 seconds for the second and third tasks. These results show that after the first task, the subjects are more comfortable in using the system.

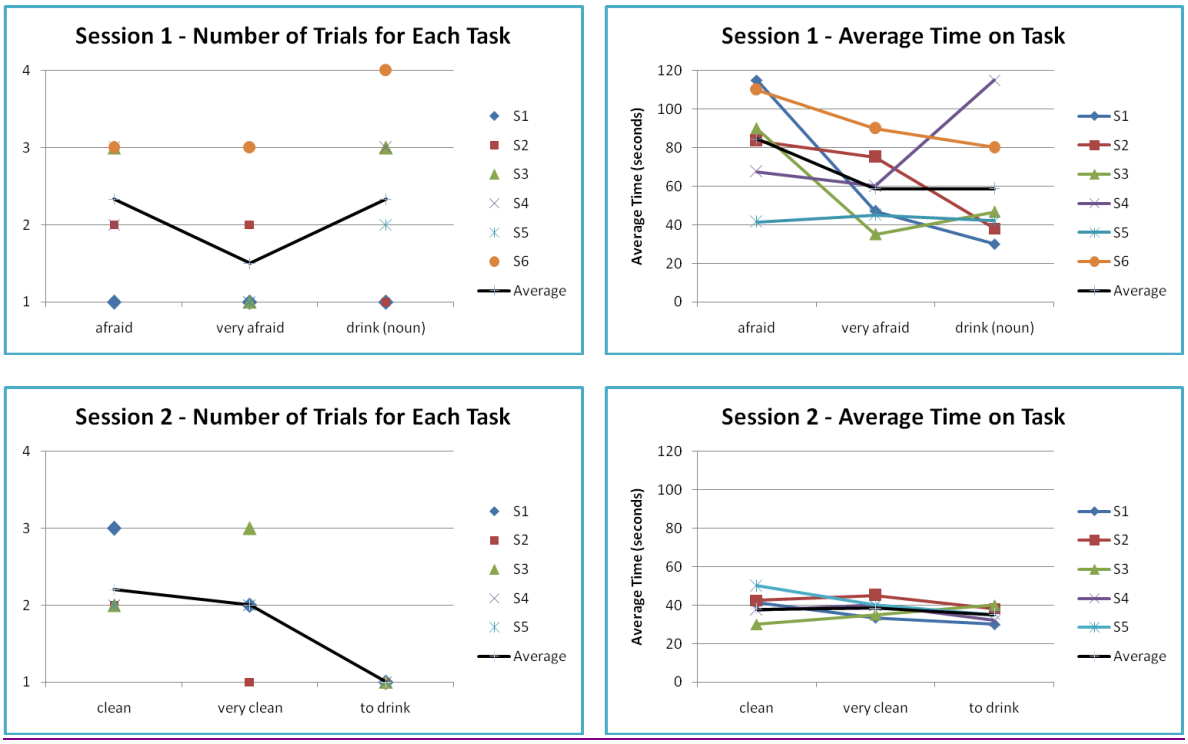


Figure 5. Task analysis for sessions 1 and 2. For each session, the number of trials and the average time on task is plotted

In session 2, the subjects are asked to practice another set of signs: CLEAN, VERY CLEAN and TO DRINK. As in the first session, the first two are variants of the same base sign, performed by two hands; and the third is a completely different sign, performed by a single hand. From the six subjects who participated in the first session, only five participated in the second one. In this session, the subjects directly started using the system without any help about the usage. The results are shown in Figure 5 and they reflect that the subjects recall the system usage, without any difficulty, and the average time on task has decreased with respect to the first session. The average number of trials is again around two for the first two signs, similar to the first session, which shows that the subjects perform a new sign correctly after two trials on average. For the third sign, all the subjects succeeded in their first trial.

During the interviews, the subjects made positive comments about the system in general. They find the interface user friendly and the system easy to use. All of the subjects indicate that using SignTutor during sign language learning can help to learn and recall the signs. The subjects find the practice part, especially its capability to analyze both hand and head gestures, very important. They commented that the possibility of watching the training

videos and their own performance together with the animation makes it easier to understand and perform the signs. They note, however, that it would be nice to have more constructive feedback about their own performance, explaining what is wrong and what should be corrected. One of the subjects found the decisions of the SignTutor very sensitive and she noted that the decision-making can be relaxed.

The subjects had no difficulty in using the glove but they commented that it would be better to be able to use the system without any gloves. One of the subjects suggested adding a picture, possibly a 3D image, of the hand shape used during the sign in addition to the sign video. This will help to understand the exact hand shape performed during the signing.

During the experiments, we were able to observe the system performance in a real life setting. A very important requirement of the system is that the camera view should contain the upper body of the user and the distance between the camera and the user should be at least one meter. We have performed the experiments in an office environment with normal lighting conditions, without using any special lighting device. The hand segmentation requires a reasonably illuminated office environment and the user's clothes should have a different color than the gloves. If these conditions are met, the system works accurately and efficiently.

V. Discussion And Conclusion

We have developed SignTutor as an application that provides a flexible, accurate and easy to use environment to assist sign language education. The tutor application is a multimodal application that analyzes the head activity and hand activity of the users, and classifies it with a sequential fusion approach. The system gives feedbacks consisting of both text information and synthesized video, which shows the user a caricaturized version of the user's signing. Our experiments show that the SignTutor system is successful in integrating non-manual signs with the manual signs and achieves accurate results. The user study has validated the usefulness and efficiency of the system in a real life setting.

The experimental results show that the non-manual features are quite important in sign language recognition if unrestricted sign recognition is intended. The usage of non-manual features increases the accuracy by around 8% in the signer-independent protocol, and 2.5% in the signer-dependent protocol. The automatic clustering and sequential fusion strategy provides a dedicated classification within the identified clusters. The fusion method assumes that the first stage classifier correctly determines the sign cluster. The uncertainty within the sign cluster is solved via the specialized second stage classifier. Although the fusion strategy is able to considerably increase the accuracy in the signer-independent protocol, there is a smaller difference in terms of accuracies in the signer dependent protocol. This is probably due to a ceiling effect where the $HMM_{M\&N}$ classifier alone achieves a very high accuracy.

There are several avenues through which this proof-of-concept tutor system can be improved. Both signer-independent and signer-dependent performance results indicate that invariance properties of the features must be further investigated. One cannot expect much aid from signer adaptation since users are novices [21]. The analysis of non-manual signals can be improved and other non-manual signals such as facial expressions, body posture can be incorporated to the system. Users can be better accommodated to the use of SignTutor by providing them with sign videos from other teachers, a text based description of the sign, picture of the hand shape(s) used during the sign, and an interactive 3D animated avatar. The latter is an advanced version of out avatar in that users can watch the sign from different viewpoints by rotating the avatar. A final point is to give more instructive feedback to trainees by giving hints about the correct position of hands with respect to the body and the other hand, in addition to the present class error message.

Acknowledgement

We have developed most of this work in a joint project during the eNTERFACE 2006 Workshop on Multimodal Interfaces, which is supported by EU FP6 Network of Excellence SIMILAR, the European taskforce creating

human-machine interfaces SIMILAR to human-human communication. This work has also been supported by TUBITAK project 107E021 and Boğaziçi University project BAP-03S106.

REFERENCES

[1] S. K. Liddell, Grammar, gesture, and meaning in American sign language, Cambridge University Press, 2003.
[2] W. C. Stokoe, "Sign Language Structure: An outline of the visual communication systems of the American deaf," Studies in Linguistics: Occasional papers 8, 1960.
[3] S.C.W. Ong, S. Ranganath. "Automatic sign language analysis: A survey and the future beyond lexical meaning", *IEEE Transactions on PAMI*, June 2005, vol.27, no.6, pp.873-891.
[4] F.K.H. Quek, "Toward a vision-based hand gesture interface", *Singh, G., S. K. Feiner, and D. Thalmann (editors), Virtual Reality Software and Technology: Proc. Of the VRST'94 Conference*, pp. 17–31, World Scientific, London, 1994.
[5] Y. Wu and T.S. Huang. "Hand modeling, analysis, and recognition for vision based human computer interaction", *IEEE Signal Processing Magazine*, 2001, v.21, p.51–60.
[6] T. Starner and A. Pentland. "Realtime American sign language recognition from video using hidden Markov models", *Technical report*, MIT Media Laboratory, 1996.
[7] C. Vogler and D. Metaxas. "ASL recognition based on a coupling between HMMs and 3D motion analysis". In *International Conference on Computer Vision* (ICCV'98), Mumbai, India, 1998.
[8] G. Fang, W. Gao and D. Zhao, "Large-vocabulary continuous sign language recognition based on transition-movement models," IEEE Transactions on Systems, Man and Cybernetics, Part A, vol. 37, no. 1, pp. 1-9, 2007.
[9] K.W. Ming and S. Ranganath, "Representations for facial expressions," Proc. of Int. Conf. on Control Automation, Robotics and Vision, vol. 2, pp. 716-721, Dec. 2002.
[10] U.M. Erdem and S. Sclaroff, , "Automatic detection of relevant head gestures in American sign language communication," Proc. Int. Conf. on Pattern Recognition, vol. 1, pp. 460-463, 2002.
[11] SignTutor demonstration video, http://www.cmpe.boun.edu.tr/pilab/pilabfiles/demos/signtutor_demo_DIVX.avi
[12] S. Jayaram, S. Schmugge, M. C. Shin, and L. V. Tsap, "Effect of color space transformation, the illuminance component, and color modeling on skin detection", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2004, pp. 813-818
[13] O. Aran, L. Akarun "Recognizing two handed gestures with generative, discriminative and ensemble methods via Fisher kernels", *International Workshop on Multimedia Content Representation, Classification and Security*, Istanbul, September 2006.
[14] Machine Perception Toolbox (MPT). http://mplab.ucsd.edu/grants/project1/free-software/MPTWebSite/API/].
[15] W. Beaudot, "The neural information processing in the vertebrate retina: A melting pot of ideas for artificial vision", *PhD Thesis in Computer Science*, INPG (France), December 1994.
[16] A. Benoit, A. Caplier, "Head nods analysis: Interpretation of non verbal communication gestures", *IEEE ICIP*, 2005, Italy
[17] A.B. Torralba, J. Herault (1999). "An efficient neuromorphic analog network for motion estimation." *IEEE Transactions on Circuits and Systems-I: Special Issue on Bio-Inspired Processors and CNNs for Vision*, February 1999, Vol 46, No. 2.
[18] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," in Proceedings of the IEEE, pp. 257--285, 1989.
[19] I.S. Pandzic, R. Forchheimer, "MPEG-4 facial animation: The standard, implementation and applications", *Wiley*, 2002.
[20] K. Balci, "Xface: Open source toolkit for creating 3D faces of an embodied conversational agent", *5th International Symposium SmartGraphics*, 2005, Germany.
[21] S. C. Ong, S. Ranganath and Y. Venkatesha, "Understanding gestures with systematic variations in movement dynamics," Pattern Recognition, vol. 39, no. 9, pp. 1633--1648, 2006.