Combined Gesture-Speech Analysis and Synthesis

Mehmet Emre Sargin, Ferda Ofli, Yelena Yasinnik, Oya Aran,
Alexey Karpov, Stephen Wilson, Yucel Yemez, Engin Erzin and A. Murat Tekalp

***Abstract*— Multi-modal speech and speaker modelling and recognition are widely accepted as vital aspects of state of the art human-machine interaction systems. While correlations between speech and lip motion as well as speech and facial expressions are widely studied, relatively little work has been done to investigate the correlations between speech and gesture.**
Detection and modelling of head, hand and arm gestures of a speaker have been studied extensively in [3]-[6] and these gestures were shown to carry linguistic information [7],[8]. A typical example is the head gesture while saying "yes". In this project, correlation between gestures and speech is investigated. Speech features are selected as Mel Frequency Cepstrum Coefficients (MFCC). Gesture features are composed of positions of hand, elbow and global motion parameters calculated across the head region. In this sense, prior to the detection of gestures, discrete symbol sets for gesture is determined manually and for each symbol, based on the calculated features, model is generated. Using these models for symbol sets, sequence of gesture features is clustered and probable gestures is detected. The correlation between gestures and speech is modelled by examining the co-occurring speech and gesture patterns. This correlation is used to fuse gesture and speech modalities for edutainment applications (i.e. video games, 3-D animations) where natural gestures of talking avatars is animated from speech.

***Index Terms*— Gesture Recognition, Keyword Spotting, Audio-Visual Correlation Analysis, Prosody Analysis, Gesture Synthesis.**

I. Introduction

The role of vision in human speech perception and processing is multi-faceted. The complementary nature of the information provided by the combinations of visual speech gestures used in phoneme production (such as lip and tongue movements) has been well researched and shown to be instinctively combined by listeners with acoustic and phonological information to correctly identify what is being said. Visual information can also provide listeners with certain aspects of paralinguistic knowledge about a speaker, helping them to be located in space, as well as supplying information regarding their age, gender and emotional intent.

The project detailed in this report, seeks to perform a preliminary exploration of potential correlations between non-facial gestures and speech, with the goal of providing natural gesture patterns for the task of artificial gesture synthesis. The project consists of a number of inter-connected modules, sketched below:

- Audio-Visual Analysis: A fine-grained examination of the audio-visual data is carried out, seeking to identify salient gesture patterns and relevant speech phenomena for potential correlation;
- Preparation of Training Data: Careful selection and preparation of samples of gestures and keywords for training automatic speech and gesture detectors;
- Head and Hand Tracking;
- Keyword Spotting;
- Gesture Recognition;
- Gesture Synthesis and Animation;

This report, as well as the source code for the software developed during the project, is available online from the eNTERFACE'05 web site: www.enterface.net.

This research was partly funded by SIMILAR, the European Network of Excellence on Multimodal Interfaces, during the eNTERFACE05 Workshop in Mons, Belgium.

II. Motivations and Initial Observations

A primary motivation of the work presented here was to identify natural classes of gestures that conveyed real linguistic meaning, that is, to identify gestures or groups of gestural patterns that could be clearly correlated with information conveyed in the speech signal. Once identified, these classes would be used to synthesize "natural" gesture patterns using an animated stick figure, given an input speech signal. The work detailed below is intended to be a preliminary investigation and so is restricted to analyzing gestures in a limited but gesture-rich task. An audio-visual database was prepared, comprising 25 minutes of video data. A single native speaker of Canadian English was recorded, providing directions to a number of known destinations in response to questions given off camera.

An initial informal analysis was carried out, in order to ascertain potential lexical candidates that had recurring patterns of significant gestures. This involved close viewing of the video data by two investigators with experience of gesture identification and speech annotation. Initial observation highlighted three candidates, "left", "right", and "straight", for further study. The three lexical items were chosen as they showed a high co-occurrence with periods of significant manual gestural activity. Furthermore, they had a high distribution throughout the database indicating a potentially rich source of data for analysis. It was informally noted that 28 instances of the candidate "left", appeared to be accompanied by some sort of gesture. Similarly 31 occurrences of "right" had accompanying gestures, while "straight" had associated gestures 32 times throughout the database. Other candidate words included "across", "no", and "down", but these were dismissed as having too few gesture-marked occurrences (8, 8, and 6 respectively).

III. Database and Tools

For the database we used a 25 minute video recording of the experiment in which a subject is asked to provide another person with directions from one location to another. The speaker is a native speaker of Canadian English and is familiar with all the locations asked.

Fig. 1. Examples of ”Straight” and ”Left” Gestures

Fig. 2. Examples of ”Nod” and ”Tilt” Gestures

A. Gesture Analysis

The video was viewed in Virtual Dub 1.6.9 which allowed the normal speed playback accompanied by sound, as well as stepping frame by frame at a rate 25 fps. Based on initial observation of directional words and gestures that were salient in the video, the following hand gestures were manually labelled:

- right and left gestures - the right or left hand turns to make a 90° angle with the arm, pointing to the right for right gesture, or to the left for left gesture;
- straight - the subject starts with her hands in parallel, palms facing each other, fingers directed up, and moves the hands away from the body by extending her elbows. The finishing position is with hands parallel, palms facing each other, fingers pointing away from the subject's body.

For each gesture identified as a right, left, or straight gesture, we noted the frame numbers corresponding to the initial and the final hand positions of the gesture movement. If the subject's hands stayed in the final hand position for several frames without movement, only the first frame was noted and recorded as the end of the gesture.

Ten examples of each gesture were snipped from the video in order to serve as training data for the gesture recognizer. Six minutes of the video were labelled for the right, left, and straight gestures without sound and used for analysis of correlation with speech which will be explained later.

Head gestures were examined and seemed to correlate with prominences in speech. Since evidence for correlation between sharp head movements and prosodic events in speech has been presented in gesture literature previously [2], we have decided to narrow down our investigation of head gestures to nods and head tilts. These gestures were manually labelled without listening to the audio and following the criteria:

- nod - the head comes down with chin closer to the body and sharply comes back up;
- tilt - the head rotates right or left 45° from its natural vertical position.

Again, ten examples of each gesture were saved as clips and served as training data for the gesture recognizer. Two minutes of the video were labelled for the nods and tilts for analysis of correlation with speech prominences.

B. Speech Analysis

The speech was investigated using the 25 minute .wav file corresponding to the video. The phonetics annotation toolkit Praat 4.3.19 was used to view the waveform, pitch, spectrogram, and intensity of the sound. All 25 minutes were manually transcribed for words ”right,” ”left,” and ”straight” using spectrogram and waveform to identify precisely beginnings and ends of these words. In the 25 minutes the word ”left” was said 28 times, the word ”right” - 29 times, ”straight” - 46, giving us a database of 103 keywords.

As we have previously mentioned, while a potential correlation between head gestures and certain lexical items may exist, (nods linked with words pertaining to agreement or assertion, tilts with words associated with hesitation), initial informal analysis, driven in part by previous research outlined in the literature [1], implied a strong correlation between head gestures and prosodic events known as pitch accents. The Tone and Break Indices (ToBI) prosody labelling convention was chosen to mark prominences in speech [19]. In order to establish an initial working hypothesis, an experienced ToBI labeller marked 2 minutes of speech for pitch accents and phrase boundaries. This two-minute section of the sound file corresponded to the video segment previously labelled for head gestures. Our labels deviated from the ToBI notational convention in that the pitch accents were marked as intervals spanning the whole accented syllable, rather than single point events. Within the 2 minute segment, there were 122 identified pitch accents.

IV. Correlation Analysis

After some manual labels have been provided for speech and gesture events, several correlation analysis were conducted in order to provide justification for two hypotheses:

- directional hand gestures are closely correlated with the identified lexical candidate tokens, such as ”left”, ”right” and ”straight”;
- sharp head movements, such as nods and tilts, are closely correlated with speech prominences marked as pitch accents.

In this section we will describe the correlation analysis procedure and results for both of these two hypotheses.

A. Directional Hand Gestures

Within the six minute video fragment labelled for directional hand gestures and speech keywords as described in the Database section, 23 gestures were manually identified and fell into categories summarized in numbers below as well as in the Figure 3 with percentages. Of the 23 gestures, 15 were direct matches with the candidate words ”left”, ”right”, and ”straight”, meaning that there was some degree of temporal

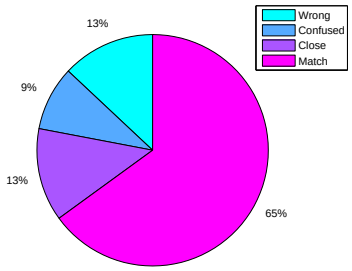

Fig. 3. Summary of Directional Word-Gesture Alignment

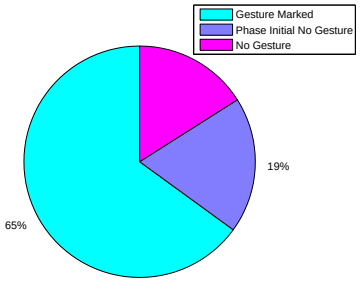

Fig. 4. Summary of Identified Pitch Accents

overlap between the gestures and corresponding keywords. A further 3 gestures were deemed to be ”close”, being in the judgement of the labeller associated with one of the candidates, but with the phase of the gesture narrowly falling outside of the duration of the word (2 were off by one frame, the third missed by 360 ms). Of the remaining 5 gestures, 3 were wrongly identified as being related and 2 were designated as ”confused”, meaning that speaker has correctly used the gesture to indicate going left, right or straight, but that the phase of the gesture overlaps with another candidate word, usually being used in a different context. For example, the phrase: ”Take a left and go *straight* down that street” had two accompanying left hand gestures. The first overlapped with the keyword ”left” and was deemed a match, the second with the keyword ”straight” and was marked as ”confused”.

B. Head Gestures

The two-minute sample file labelled for prosody and sharp head movements was found to contain 122 pitch accents and 81 head gestures 66 nods and 15 tilts. Of the 122 pitch accents, 79 or 64.75% overlapped with a head gesture, either a nod or a tilt. It is worth noting, that from the 43 pitch accents that did not overlap with a head gesture, 23 or 53.5% were phrase initial accents, which are known to be problematic in prosody labelling (see Figure 4. Often phrase initial stressed syllables are misidentified as pitch accents due to the fact that both pitch accents and phrase initial syllables are accompanied by tense voice quality [20].

If we disregard the 23 phrase initial syllables that were labelled as accents, only 20 of the 100 pitch accents identified

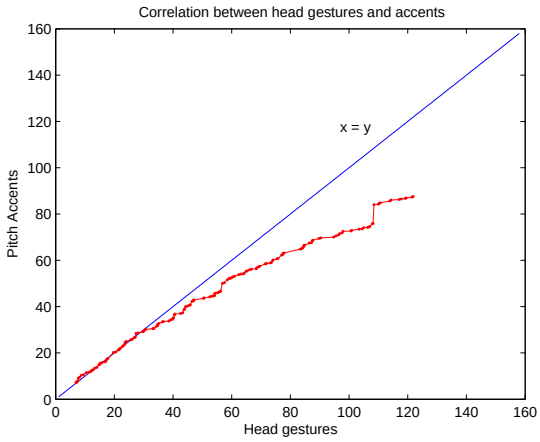

Fig. 5. Head Gesture and Accent Correlation Graph

did not overlap with a sharp head movement, that is 80% of remaining accents co-occurred with a head gesture.

The 79 accents that overlapped with a nod or a tilt were also examined for temporal correlation with the relevant head gesture. Time-stamp labels of the accented syllable were compared to the start and end time-stamps of the overlapping gesture using the statistical test of Pearson’s correlation ran in Matlab. The correlation test produced Pearson’s correlation coefficient r=0.994, which implies almost perfect correlation. The corresponding correlation plot can be seen in figure 5.

V. Recognition of Auditory Events

The ultimate aim of the recognition process described in the following section is to automatically detect selected auditory events, namely the keywords, ”left”, ”right”, and ”straight”, as well as pitch accents. Detected instances of the chosen events will act as cues to animate the stick figure with correlated gestures.

A. Manual Labelling

In order to supply high-quality training data for the automatic keyword detector (Section V-B.1), the labels of the three keywords for a 20 minute portion of the sound file were used. Since a silence detector was also being trained, a number of examples of silence were also identified and labelled. All remaining non-keyword and non-silence portions of the 20 minute segment were presumed to be ”garbage”.

B. Automatic Labelling

Unlike in the keyword spotting procedure, the manual prosodic labels for the two-minute segment in the database were not intended to act as a training set for an automatic detector. Instead, they served as a ”gold-standard” against which we could measure the effectiveness and accuracy of our automatic pitch accent detection strategies.

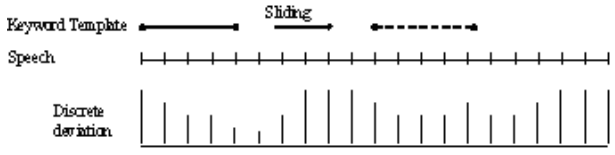

Fig. 6. Obtaining acoustic deviation function for a keyword

1) Keyword Spotting: The part of research during the project was connected with realization of special speech recognition system (keyword spotting system) for finding in unknown speech the words which describe the direction: left, right and straight. In our case it should be speaker-dependent because we have only one voice. Such system can be based on known techniques for speech recognition: dynamic time warping or statistical modelling.

a) Dynamic time warping based keyword spotting system: Dynamic time warping method is the approach, which allows finding an optimal match between two given sequences (e.g. time series). The dynamic programming algorithm is usually used for searching the optimal match. This method was firstly applied for automatic speech recognition in the 60's for isolated word recognition. Among the advantages of this method the following can be selected: easy realization, the stage of training of acoustical models is not required as well as any necessity to prepare the training speech data. During the eNTERFACE'05 workshop the original method for keyword spotting was realized using C++. This method uses analysis in sliding window for comparison of keyword template with fragment of speech with calculation of acoustic deviation function along the speech utterance for each keyword [10]. The keyword template is the most typical pronunciation of keyword by the speaker. Also several pronunciations of each keyword can be used for analysis of the speech. The input signal from wave file enters into the module of parametrical speech presentation. In this module the sequence of digital samples is divided into speech segments. And a vector of parameters is calculated for each such segment. For parametrical representation we used the Mel Frequency Cepstral Coefficients (MFCC). The calculation of speech parameters is fulfilled by the parametrization module HCopy included in Hidden Markov Toolkit (HTK). Then the each parameterized keyword template (”LEFT”, ”RIGHT”, ”STRAIGHT”) is shifted (slide) along the speech signal with some step and keyword template is compared with fragment of speech of the same length by dynamic programming (DP) method with calculation of deviation estimation between template and speech fragment in sliding window. Figure 6 shows the process of sliding DP-analysis. The word template slides along input signal with slide step and the DP-deviation between the template and signal part is calculated for every step forming the discrete deviation function. The smaller DP-deviation between the keyword template and signal part the more probability of this keyword appearance. At that the sliding step from 1 segment of speech till several speech segments can be applied. To select hypothesis of keywords in three streams of deviation functions (Figure 7) we use the thresholds which depend on the length

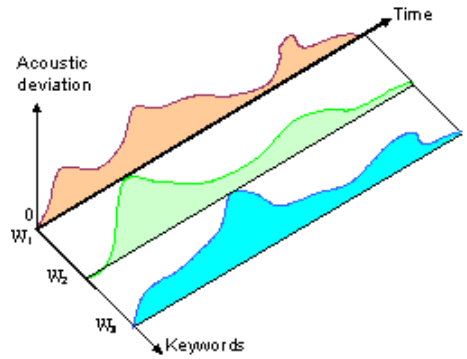

Fig. 7. Sliding analysis in the system

Recognized keywords	Missed words	False alarms
33	2	22

TABLE I
KEYWORD SPOTTING PERFORMANCE USING SLIDING ANALYSIS

of each template. Changing the thresholds we can manage the performance of the system and find some balance between word detection accuracy and number of false alarms made by the system. On the output of sliding analysis algorithm we combine the outputs of each stream and form the time-stamps for keywords in analyzable speech. For testing the system we used speech of one speaker with duration 330 seconds (5,5 minutes). In this speech there exist 35 keywords and this fragment contains about 600-700 continuously pronounced words. This fragment also was manually labelled, but it is required only for evaluation of the results of the system. The results of usage of this method are presented in Table I.

We used three criteria for evaluating the system: amount of properly recognized keywords in test speech, amount of missed keywords in test speech and amount of false alarms at analysis of speech. Thus the method showed 94,3%(33 of 35) in accuracy of keyword spotting and the same time gives about 3,6% (22 of 600) false alarms during analysis the speech. These results are not well enough but taking into account that the method does not require the construction and training of the models of words it can be used in some application areas for keyword spotting task.

b) Hidden Markov Model based keyword spotting system: At present the Hidden Markov Models (HMM) are most popular technology for statistical speech modelling and processing for diverse domains (not only for speech processing). It is difficult to realize effective system for statistical modelling using HMM is short time and therefore we used free available toolkit for training the HMMs. As the base technology for development of keyword spotter we have chosen the Hidden Markov Toolkit (HTK) developed in Cambridge University Engineering Department. HTK is free available toolkit which can be downloaded in Internet [11] and source code in C is available. Among the advantages of HTK the following should be noted:

- World recognized state-of-the-art speech recognition system

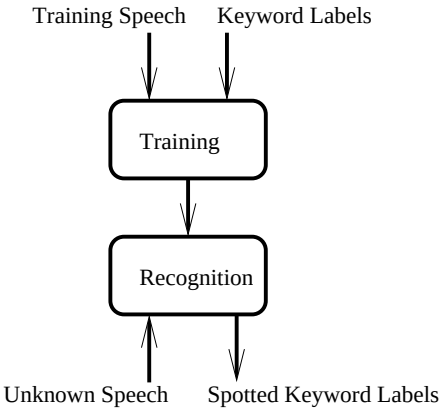

Fig. 8. General structure of system using HTK

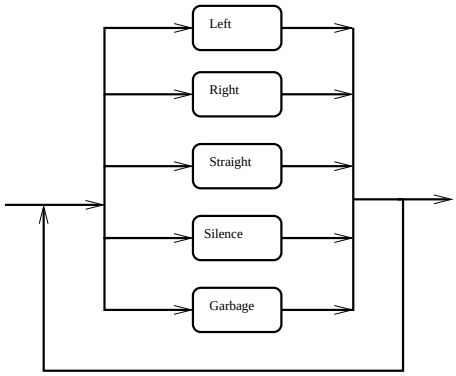

Fig. 9. Task Grammar

- Support a variety of different input audio file formats
- Support different feature sets
- Support almost all common speech recognition technologies.

The modelling of speech by HMM includes two stages (Figure 8):

1) Training the HMM using the database of training speech which was manually labelled
2) Testing this system by test speech.

The stage of models training includes the following steps:

- Definition of dictionary (lexicon of the task)
- Definition of task grammar - Preparing training speech data
- Coding the speech data (feature extraction)
- Definition of topology of HMMs (prototypes)
- Creating initial HMM models
- Re-estimation of HMMs parameters using speech data
- Mixture Splitting

At first the grammar for speech recognition should be defined. We used HMM for three keywords: ”LEFT”, ”RIGHT” and ”STRAIGHT” as well as defined the models for ”SILENCE” (it is signal without any speech but with background noise only) and for ”GARBAGE” (it is any other speech). This approach is similar to the approach described in [10]. The grammar for our task is shown on the Figure 9.

20 minutes of manually labelled speech was used for training the keyword spotting. Each keyword was pronounced in training speech at least 30 times. The labels ”left”, ”right”, ”straight” and ”silence” with corresponding time of beginning and time of ending were set during manual analysis of training speech. The labelling was made using software Praat 4.3 but the output format of Praat with labels was not suitable directly for HTK processing because Praat uses the time-stamps in seconds but HTK requires timestamps in 100 ns items. The special software was developed during the project for converting labelling data from the Praat format into HTK format. The feature extraction was performed using HTK configured to automatically convert input Wav files into vectors of parameters. As set of features we used Mel Frequency Cepstral Coefficients + delta coefficients + acceleration coefficients. Each feature vector includes 39 components: MFCC - 13, delta coefficients - 13, acceleration coefficients - 13. The next step in training stage was the definition of prototypes for HMMs. The parameters of prototype are not important, its purpose is to define the model topology. We used the left-right HMM with continuous observation densities in HMM. The number of states for each keyword depended on the number of real phonemes in each word. Thus for ”LEFT” and ”RIGHT” we used 14 states and for ”STRAIGHT” - 20 states. This amount is calculated as number of phonemes in pronunciation of keyword multiplied by 3 and plus 2 states intended for concatenation of models. The prototypes for ”SILENCE” and ”GARBAGE” include 5 states each. Then the initial HMM models were created using the training speech and labels manually made in this speech. The re-estimation of parameters was performed using standard Baum-Welch algorithm for continuous density HMMs. After this step we obtained the trained HMMs for all words in our task. For recognition and testing the system we used 5,5 minutes of other speech of the same speaker. The recognition is performed by the Viterbi algorithm, the usage of N-best list on the output of the algorithm in this task is not possible. Thus we analyzed only an optimal hypothesis of speech recognition. According to the first experiments keyword spotter was able to find almost all keywords in test speech, but it gave many false alarms (above 30%). To decrease the number of false alarms we tried to apply the threshold for acoustical estimates but it did not give the acceptable results. The best results of system performance were obtained using mixture splitting and applying the multi Gaussian HMMs. In HTK the conversion from single Gaussian HMMs to multiple mixture component HMMs is usually one of the final steps in building a system. It allows creating more precise models for available training data. The number of mixture components in HMMs is repeatedly increased until achievement of the desired level of performance (keyword spotting accuracy and amount of false alarms).

The Figure 10 shows the dependence of keyword spotting accuracy and inverse rate of false alarms (100% - % of false alarms in speech). In can be seen that increasing the number of Mixtures of Gaussians we decrease the number of false alarms, because we tune our models better to the available training data and create more precise HMMs. But after some point the keyword spotting accuracy are decreased. It can be explained by amount of available pronunciations of keywords in training speech. Of course using other set of training speech

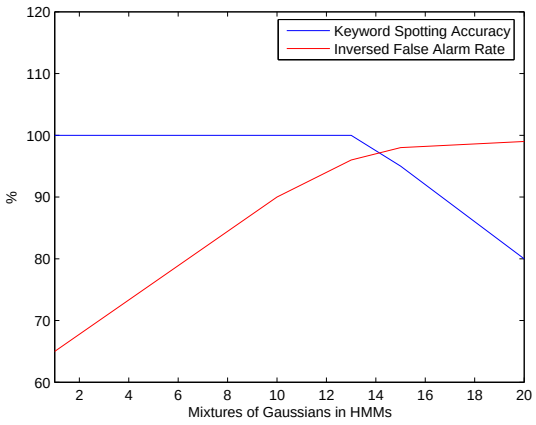

Fig. 10. Dependence of keyword spotting accuracy and inversed false alarm rate from the number of mixtures of Gaussian distribution in HMMs

Recognized keywords	Missed words	False alarms
33	2	10

TABLE II
KEYWORD SPOTTING PERFORMANCE USING HIDDEN MARKOV MODELS

these results will be changed.

Thus the best results on system performance were achieved by using 10-15 Gaussian mixtures in models. Table II shows the best results of keyword spotting by Hidden Markov Models based approach where the balance between keyword spotting accuracy and false alarm rate was found.

These results mean that we have 94,3% (33 of 35) in accuracy of keyword spotting and 1,6% (10 of 600) false alarms during analysis the test speech. These false alarms can be partly explained by specificity of pronunciation of the speaker. For instance sometimes in continuous real speech she pronounced the keyword ”STRAIGHT” as ”s t r i t” that is the same pronunciation as for the out-of-vocabulary word ”STREET” or in her speech the keyword ”RIGHT” was pronounced like ”r e i”. Thus after comparison of the Table I and Table II we have chosen the second version of keyword spotter based on Hidden Markov Models for the joint multimodal system in the project. The methods showed almost the same keyword spotting accuracy but the second approach was better in such criterion as false alarm rate.

2) Prosodic Event Spotting: Motivated from the correlation between accents and head movements, we propose an automatic methodology to extract accents from the speech signal. Proposed methodology uses pitch contour and intensity values as features. Pitch contour and intensity values have a frame rate of 100 samples per second. Pitch contour is extracted from the speech signal using autocorrelation method described in [9]. In order to extract intensity values of speech signal, the values in the sound are first squared, then convolved with a Kaiser-20 window with side-lobes below -190 dB. The effective duration of this analysis window is $3.2/(100Hz)$, where $100Hz$ is selected as minimum pitch frequency. The duration of analysis window should guarantee that a periodic signal is analyzed with a pitch-synchronous intensity ripple not greater than our 4-byte floating-point precision.

Recognized Accents	Missed Accents	False Alarms
68%	32%	25%

TABLE III
ACCENT DETECTION PERFORMANCE

Algorithm first detects high intensity speech regions which are above the threshold $t_s = 48dB$. Median filter is used to smooth out small peaks from detected speech regions. Connected component analysis is applied to output of median filter in order to extract significantly long accent candidate regions. For each accent candidate regions, related pitch frequency sequence is investigated to eliminate non-accent regions. Number of peaks in the pitch contour for non-accent regions are usually few (0-2) or many (8-15). Therefore, the accent candidate regions that contain few or many peaks are eliminated and remaining regions are selected as accent regions.

The performance of proposed accent detector is determined using first 2 minutes of the database. Accent detector detected 85 accents out of 125 and the number of false alarms are 32. Performance rates of the accent detector can be seen in Table III.

VI. RECOGNITION OF GESTURAL EVENTS

In this section we present a framework for gestural event detection. Proposed framework can be divided into three tasks:

1) Manual Labelling of Gestures
2) Automatic Recognition of Head Gestures
3) Automatic Recognition of Hand Gestures

In the recognition phase, HMM based gesture recognizer is used and the HMM for each gestural event is trained using the manually labelled gestures.

A. Manual Labelling of Gestures

In order to train the automatic gesture detector and hand motion modeler, proper 10 examples for each gesture are labelled manually. Since all of the gestures are not well prepared gestures, elimination of non proper gestures is necessary.

B. Automatic Recognition of Head Gestures

In this section we present a methodology for head gesture recognition. Proposed methodology consists of three main tasks which are tracking of head region, extraction of head gesture features and recognition of head gestures based on Hidden Markov Models (HMM). Optical flow vectors calculated on head region are used to estimate new head position. New estimate of head region is corrected using skin color information. Head gesture features are extracted by fitting global head motion parameters to optical flow vectors. HMM is applied for recognition of gestures given the gesture features.

Fig. 11. Initial Head Region

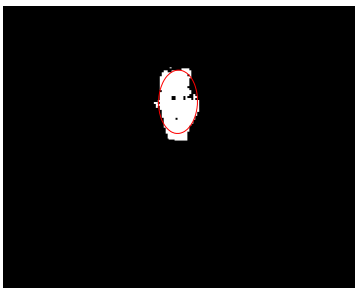

Fig. 12. Skin Region and Fitted Ellipsoid

1) Initialization of Head Tracker: Consider the first frame of the frame sequences. Since we do not have any prior knowledge about the initial position of head in the image, one should exhaustively search for face in the initial frame. For this purpose, boosted Haar based cascade classifier structure is used. Proposed object detector has been initially proposed by Viola [13] and improved by Lienhart [14]. The classifier is trained with positive and negative examples which are a few hundreds of sample views of face with size MxN and arbitrary images of the same size respectively.

Classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. Classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques which are Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost.

Trained classifier can be applied to a test image of the same size to determine whether applied image shows training object or not. One can find training objects of the same size on a whole image by running classifier on overlapping search windows across the image. In order to find same object with different sizes in other words to make classifier scale invariant, whole image can be scanned with different sized classifiers. Note that, once the classifier is trained using a specific sized object, the size of classifier is easily resized without training an another classifier with different sized objects. Sample initial head position found by boosted Haar based cascade classifier can be seen in Figure 11.

2) Extraction of Skin Blobs: Skin blobs are extracted using color information. Here we assumed that distribution of Cr and Cb channel intensity values which belong to skin regions is normal. Thus the discrimination function can be defined as Mahalonobis distance from sample point $\mathbf{x}$ to (μ_t, Σ_t). Where $\mathbf{x}$ is vector containing Cr and Cb channel intensity values $[x_{Cr}, x_{Cb}]^T$ and μ_t and Σ_t are the mean and variance of training skin regions respectively. Sample skin blob extracted can be seen as white ares in Figure 12.

3) Fitting Ellipsoid to Skin Blob Boundaries: Suppose there are m points on the contour of a skin blob. Let $\mathbf{B} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m]$ where $\mathbf{x}_i = [x_{1,i}, x_{2,i}]^T$. Then center of the ellipsoid is given by $\mu_e = E\{B\}$. Principal axis and length of each principle axis will be the eigenvector and square root of eigenvalues of $\mathbf{B}\mathbf{B}^T$ respectively. Ellipsoid fitted to skin blob can be seen in Figure 12.

4) Optical Flow Observations: Optical flow vectors will be calculated on the initial head region which was obtained in the previous step. Most trackable n points are selected on the initial head region by considering cornerness measure. Optical flow vectors are calculated on these n points which have the highest cornerness measure. Cornerness measure ρ can be defined as the minimum eigenvalue of covariance matrix of derivative image over the neighborhood S.

$$M = \begin{bmatrix} \sum_S \left(\frac{\partial I(x_1,x_2)}{\partial x_1}\right)^2 & \sum_S \frac{\partial I(x_1,x_1)}{\partial x_2}\frac{\partial I(x_1,x_2)}{\partial x_2} \\ \sum_S \frac{\partial I(x_1,x_1)}{\partial x_2}\frac{\partial I(x_1,x_2)}{\partial x_2} & \sum_S \left(\frac{\partial I(x_1,x_2)}{\partial x_2}\right)^2 \end{bmatrix} \quad (1)$$

$$\rho = \min(\text{eigval}(\text{cov}(M))) \quad (2)$$

Hierarchical Lukas-Kanade technique is applied to find motion vectors on n points. Implementation and technical details of algorithm can be found on [15].

5) Estimation of Head Position: Once the optical flow vectors are obtained on n points global motion parameters are fitted to n optical flow vectors calculated at n points. Estimation of global head motion parameters are given in Section VI-B.6. Center of search window is warped using global head motion parameters. Where warped search window will be the estimated position of head region in the next frame.

6) Head Motion Feature Extraction: Let optical flow vectors calculated at n points $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$ are represented with $\mathbf{d}_i = [d_{1,i}, d_{2,i}]^T$ where $\mathbf{x}_i = [x_{1,i}, x_{2,i}]^T$. Global motion parameters $[a_1, a_2, \ldots, a_8]$ should satisfy the equation:

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & 0 & 0 & 0 & -x_{11}^2 & -x_{11}x_{21} \\ 0 & 0 & 0 & 1 & x_{11} & x_{21} & -x_{11}x_{21} & -x_{21}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & 0 & 0 & 0 & -x_{1n}^2 & -x_{1n}x_{2n} \\ 0 & 0 & 0 & 1 & x_{1n} & x_{2n} & -x_{1n}x_{2n} & -x_{2n}^2 \end{bmatrix} \quad (3)$$

$$a = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_7 \\ a_8 \end{bmatrix} \quad and \quad d = \begin{bmatrix} d_{1,1} \\ d_{2,1} \\ \vdots \\ d_{1,n} \\ d_{2,n} \end{bmatrix} \quad (4)$$

$$\mathbf{X}\mathbf{a} = \mathbf{d} \quad (5)$$

Since $\mathbf{X}$ is tall system is overdetermined. Therefore one can find the solution using least squares and the least squares solution is given by:

$$\tilde{\mathbf{a}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{d}. \quad (6)$$

Note that, origin is selected as image center because the matrix $(\mathbf{X}^T\mathbf{X})$ is rank deficient if we select image origin as upper left corner. Note also that the more region of interest is far away from origin, the more the condition number is.

7) Head Gesture Recognition and Gesture Spotting: During the initial observation of the database, two head gestures (*down* and *tilt*) are identified as correlated with the pitch accents. The purpose of head gesture recognition in this project is to automatically detect the head gestures in the whole video. Since the number of head gestures are high and manual detection for the whole video is hard and time consuming, automatic detection is preferred. The automatically detected head gestures are used in the gesture-speech correlation analysis. The whole training set for the head gestures is formed from the first 20 minutes of the 25 minute video by manually snipping 14 down gestures and 16 tilt gestures. The features used in the training set are the 2D coordinates of the center of the head and 8 global motion parameters, extracted using optical flow vectors for each frame of the gesture sequence.

We have trained a left-to-right Hidden Markov Model for each gesture and applied an isolated gesture recognition. For measuring the performance, we used only 10 down gestures and 11 tilt gestures and reserved 4 down gestures and 5 tilt gestures for testing. For a given hand trajectory, each gesture model is tested and the one with the maximum likelihood is selected. The average recognition accuracy is given in Table IV. Results are given for HMMs with 5 states and with 5 mixtures of Gaussian.

Gesture	Accuracy on Training Set	Accuracy on Test Set
down	0.883	0.875
tilt	1	1

TABLE IV
ISOLATED HEAD GESTURE RECOGNITION PERFORMANCE

To automatically detect the gestures in a video stream, a gesture spotting methodology must be used. In speech, for keyword spotting, a garbage model is formed as well as the keyword models. The garbage model for speech can be trained with clearly defined non-keywords. However, in gesture spotting, it is not clear what a non-gesture is. To overcome this problem, Lee and Kim [17] proposed a threshold model that utilizes the internal segmentation property of the HMM. For gesture spotting, we have used the approach of threshold model of Lee and Kim. The threshold model is formed by using the states of the gesture models. All outgoing transitions of states are removed and all the states are fully connected such that in the new model, each state can reach all other states in a single transition. Prior probabilities, observation probabilities and self-transition probabilities of each state remain the same, and probabilities of outgoing transitions are equally re-assigned. For a particular sequence to be recognized as a gesture, its likelihood should exceed that of the other gesture models and the threshold model.

Once the threshold model is formed, gesture spotting can be performed. When continuous stream is given as input, we start with the first frame and increase the sequence length until we identified the sequence as a gesture. After a gesture is identified, spotting continues with the next frame after the end of the gesture. To speed up the process, minimum and maximum lengths for a gesture can be used.

We applied gesture spotting on the 2 minute data. Figure 13 shows the correlation between head gestures and accents in speech. Also the overlap between the manually labelled and automatically spotted head gestures can be seen. When compared to manual labelling, automatic spotting finds *down* gestures more frequently and *tilt* gestures less frequently.

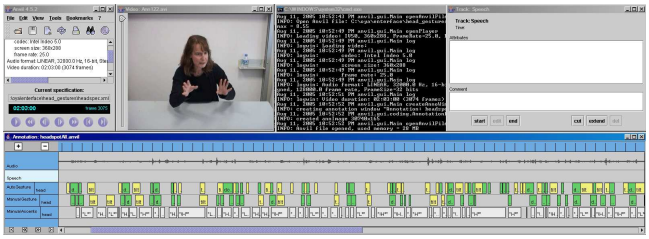

Fig. 13. Correlation between accents and head gestures. Accents are manually labelled. Gestures are both manually labelled and automatically spotted

C. Automatic Recognition of Hand Gestures

In this section we present a framework for hand gesture recognition. Proposed framework consists of three main tasks which are tracking of hand region, extraction of head gesture features and recognition of head gestures based on Hidden Markov Models (HMM). Center of mass position and velocity of each hand is tracked and smoothed using two different filters which are Kalman and Particle Filter. Smoothed position and velocity of center of mass is defined as hand gesture features. HMM is applied for recognition of gestures given the gesture features.

1) Initialization of Hand Tracker: In order to extract initial position of hand regions one should exhaustively search for hand in the initial frame. We propose two methods for determination of hand regions in the initial frames.

First methodology is based on skin color information. Given an initial frame, skin colored regions are extracted using the methodology described in Section VI-B.2. However, in this case, unlike the head region correction algorithm, region of interest is selected as whole image. Thus the skin regions coming from head are also marked as candidate hand regions. After obtaining all hand region candidates, connected component analysis is applied to determine connected regions in the set of hand region candidate pixels. The connected components that are larger than or smaller than thresholds t_h and t_s are discarded. In our experiments t_{high} and t_{low} are selected as 100 and 300 pixels. Remaining skin colors are detected as hand region using a semi-automatic method in which software asks user for verification of hand regions.

Second methodology is based on boosted Haar based cascade classifiers. In addition to face detection task VI-B.1 Boosted Haar based cascade classifiers can also be applied to detect hand regions. However, unlike face detection, there is no common classifier structure for hand detection task. Therefore, hand detector classifier is trained using 240 sample hand

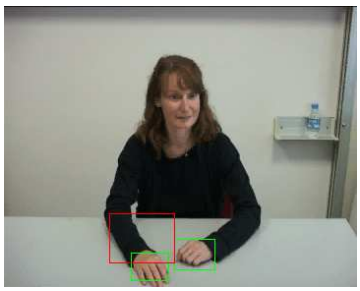

Fig. 14. Initial Hand Regions

regions which are selected from "our database". Recognition performance of Haar based classifier is high however false alarm rate is not as good as Haar based classifier for faces. As an expected result, there is a tradeoff between recognition performance and false alarm rate. In order to detect most of the poses correctly one should train the system using the training samples that contains significant number of poses. This results in an increase in the false accept rate. In contrast if one trains the classifier with specific pose only, the recognition performance of the classifier drops significantly. Therefore, detection and false alarm rate for Haar based classifiers will be high in the task of recognition of hands in specific pose like sign language. In order to decrease the false alarm rate, skin color information is fused with Haar based classifier decisions. Fusion rule is looking at he hand region candidates and check weather there is significant amount of skin colored pixels in candidate region. The threshold t_h for determination of hand region is 10% where the Haar based classifier decision is a bounding box for hand. Sample initial hand positions found by boosted Haar based cascade classifier can be seen as green and red rectangles in Figure 14 where red rectangle is eliminated by using the method described above.

2) State-Space Model for Kalman Filtering: Kalman filter based state-space estimator assumes that the motion of a pixel can be approximated using the motion model

$$\begin{array}{rcl} x_t & = & x_{t-1} + v_{x,t-1}T + a_{x,t-1}T^2/2 \\ y_t & = & y_{t-1} + v_{y,t-1}T + a_{y,t-1}T^2/2 \\ v_{x,t-1} & = & v_{x,t-1} + a_{x,t-1}T \\ v_{y,t-1} & = & v_{y,t-1} + a_{y,t-1}T \end{array} . \quad (7)$$

Here, T denotes the frame capture rate of acquisition system which is $1/25$ fps. Velocity and acceleration in each direction x, y is represented with v_x, v_y and a_x, a_y respectively.
Moreover, motion model defined in 7 can further be approximated by ignoring acceleration term since we can neglect the change in acceleration between two consecutive frames. Neglecting the acceleration has another advantage that each derivative operation is corrupted by noise and simply discarding the higher order derivatives yield better system performance if we consider noise/precision tradeoff. Thus the state motion model becomes

$$\begin{array}{rcl} x_t & = & x_{t-1} + v_{x,t-1}T \\ y_t & = & y_{t-1} + v_{y,t-1}T \end{array} . \quad (8)$$

Let the position of center of mass of hand is x, y. The motion of each pixel in two dimensions can be approximated using model 8:

$$x_t = x_{t-1} + v_{x,t-1}T \quad (9)$$

$$y_t = y_{t-1} + v_{y,t-1}T \quad (10)$$

Thus the motion model motivates the following state-space model with state s and observations z:

$$\begin{array}{rcl} s_{t+1} & = & Fs_t + Gu_t \\ z_t & = & Hs_t + v_t \end{array} \quad (11)$$

$$s_t = [x_t, y_t, v_{x,t}, v_{y,t}]^T \quad (12)$$

$$z_t = [x_t, y_t]^T \quad (13)$$

$$F = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (15)$$

Process noise u_t and the measurement noise v_t are assumed to be uncorrelated, with zero-mean white gaussian distributions and corresponding covariance matrices Q and R. Q and R matrices are adjusted by using method [16] which tests the whiteness of the innovations process.

3) Hand Tracking with Kalman Filter: After obtaining the initial hand regions for each hand, we define two search windows based on the bounding box dimensions provided by the hand detector. Two independent Kalman filters are initialized using the positions provided by the hand detector. For each iteration, Kalman filter time update equations are calculated to predict the new hand position. The predicted hand position is used to warp search window. Mean of skin color pixel positions inside the search window is calculated and and provided to Kalman filter as the observation. Kalman filter measurement update equations are calculated to correct using the observations. Posterior states of each Kalman filter is defined as feature vectors.

4) Recognition of Hand Gestures: The hand gestures that are identified are the gestures performed when the keywords are spoken, so there are three hand gestures related with the three keywords: *left*, *right* and *straight*. The purpose is to build good models of each hand gesture so that the models can be used for the animation part. Recognition performance is used to check the quality of the produced model. The whole training set for the hand gestures is formed from the first 20 minutes of the 25 minute video by snipping the parts that corresponds to manually labelled keywords. We manually checked and eliminated some videos where there is no meaningful gesture even there is a keyword. The final training set contains 20 left, 24 right and 28 straight gestures. The features used in the training set are the 2D coordinates of the center of the left and right hands and their velocities for each frame of the gesture sequence.

We have trained a left-to-right Hidden Markov Model for each gesture and applied an isolated gesture recognition. For measuring the performance, we used 30% of the data for testing. For a given hand trajectory, each gesture model is tested and the one with the maximum likelihood is selected. The average recognition accuracies are given in Tables V, VI, VII. We trained different models for right hand, left hand and using both hands. Each HMM is trained with 5 states and with 1 Gaussian mixture. Although the recognition rates on test set are low, training set accuracies are high. This result indicates that the generalization ability of the learned models are poor but the models are good at creating the training data. Therefore, these HMM models can be used for animation purposes rather than recognition purposes. The technique used for restoring the hand trajectory using the produced models are given in detail in the Animation section.

Gesture	Accuracy on Training Set	Accuracy on Test Set
left	0.85	0.66
right	1	1
straight	0.8	0.25

TABLE V
ISOLATED HAND GESTURE RECOGNITION PERFORMANCE – USING ONLY RIGHT HAND

Gesture	Accuracy on Training Set	Accuracy on Test Set
left	0.92	0.7
right	0.76	0.2
straight	0.8	0.5

TABLE VI
ISOLATED HAND GESTURE RECOGNITION PERFORMANCE – USING ONLY LEFT HAND

Gesture	Accuracy on Training Set	Accuracy on Test Set
left	1	0.83
right	1	0.71
straight	1	0.70

TABLE VII
ISOLATED HAND GESTURE RECOGNITION PERFORMANCE – USING BOTH HANDS

VII. ANIMATION

A. Stick Model and 3D Body Model

Given a speech sequence, keyword spotter described in Section V-B.1 and accent detector described in Section V-B.2 are used to extract time-stamps of auditory events. These time-stamps and speech sequence are provided to animation engine to animate the virtual body. Initially virtual body is at the stable state and for each frame, animation engine checks for the time-stamps. If there is a coincidence between time-stamps and frame-stamps, corresponding body part is actuated to be at a moving state. In this project, we realized two animation schemes:

1) Stick Model: Stick Model consists of line segments that corresponds to forearm and upper arm where starting and ending points of these line segments are determined as hand, shoulder and elbow positions. Together with these line segments head is included with a line segment between head position and the center of the line segment between left and right shoulder. Animation engine for Stick Model uses 2D coordinates of the corresponding points.

2) 3D Body Model: 3D Body Model consists of 2 arms and head without the body. Animation engine for this model uses a dictionary of gestural events and frames are constructed manually for each event in the dictionary. Animation engine uses each event independently for the animation of head, left arm and right arm.

B. Head and Hand Motion Models

In order to animate the body model, the center of mass positions of head and both hands is required by the animation engine. For each acoustical event, related gesture synthesized by considering the duration of acoustical event and the previously recognized gestures.

1) Hand Motion Model: Figure 15 shows the trajectories of left and right hand for the hand gesture examples in the training set. Each trajectory is shifted such that the origin is (0,0). For the left gesture, the motion of the right hand is limited when compared to the motion of the left hand. Similarly for the right gesture, the motion of the left hand is limited when compared to the motion of the right hand. However for the straight gesture, both hands have large trajectories. The hand models for each hand gesture are constructed by HMMs. For the left gesture, we trained an HMM by using only the left hand trajectory, for the right gesture, we trained an HMM by using only right hand trajectory and for the straight gesture we trained two HMMs: one for the left hand and one for the right hand.

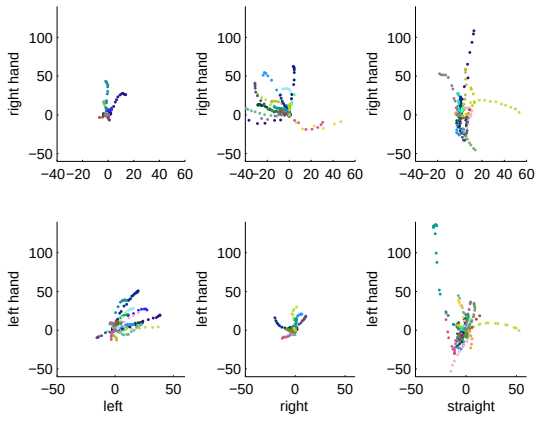

Fig. 15. Left and right hand trajectories of the gestures

To construct an observation sequence from the HMM models, we used the model parameters: state transaction probabilities, parameters of gaussian distribution (covariance matrix

and mean for the feature vector) for each state and prior probabilities of states (since HMMs are left to right, always start with the first state). Using these information, we can construct an observation sequence by just providing a sequence length. To have an idea about the sequence lengths (number of frames) of the hand gestures, we first draw the histogram of sequence lengths and then applied a normality test. Figure 16 shows the histograms and plots of the normality tests for each gesture. The test results show that the distribution of sequence lengths for each gesture is close to normal distribution. If there are no other information about the sequence length when constructing an observation sequence, a random length can be selected from the related distribution. Table VIII shows the normal distribution parameters for gesture types.

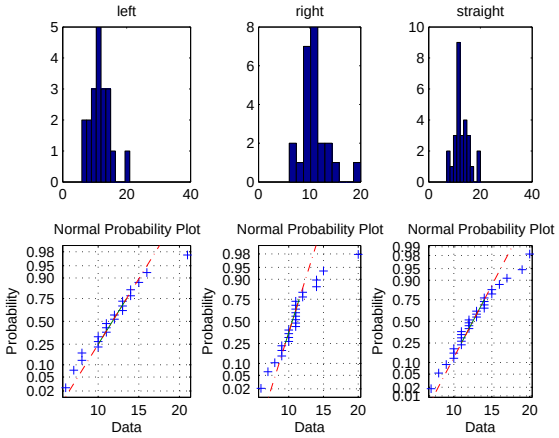

Fig. 16. Sequence length histograms and normality test plots

Gesture	μ	σ
left	11.75	3.43
right	10.96	2.82
straight	12.64	3.02

TABLE VIII
NORMAL DISTRIBUTION PARAMETERS FOR THE GESTURE LENGTHS

The methodology used for constructing the observation sequence, given a sequence length and model parameters is as follows:

1) Generate random numbers between [0,1) for each observation (so the number of random numbers generated must be equal to the sequence length)
2) Using the prior probabilities and the first random number, decide the first state (e.g. If there are 3 states with prior probabilities 0.2, 0.5, 0.3, then the decision is given by first taking a cumulative sum of the probabilities, which is 0.2, 0.7, 1. If the generated random number is less than 0.2, then select state 1 as the first state. If the random number is between 0.2 and 0.7, select the second state and if it is higher than 0.7, select the third state.
3) For i=2:sequence length
4) Decide i^{th} state using the state transition probabilities, the previous state and the i^{th} random number
5) End for
6) For i=1:sequence length
7) Construct the i^{th} observation by producing a random number from the gaussian distribution of the i^{th} state
8) End for

By using this methodology, we produced hand trajectories for each gesture where, for the *left* gesture, only left hand moves; for the *right* gesture, only right hand moves; and for the *straight* gesture both hands move.

On the last 5 minutes of the database, we first run the keyword spotting algorithm for finding the time-stamps for words *left*, *right* and *straight*. We then produced the related hand gestures which are animated during the same period with the keyword. The sequence length is determined by the length of the keyword. Figure 17 shows the produced trajectories. As seen from the figure, left gestures are aiming left and right gestures are aiming right and straight gestures move in the y direction. This plot is similar to the one of the training data (Figure 15).

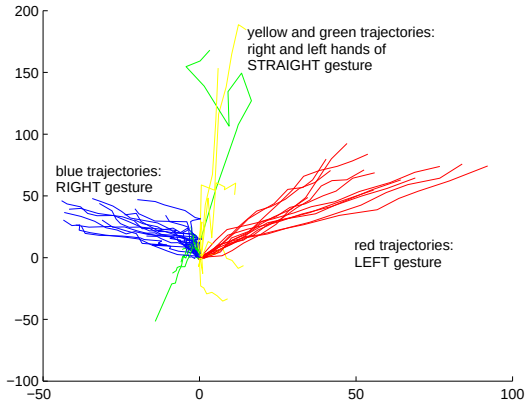

Fig. 17. Hand trajectories of the produced gestures

2) Head Motion Model: Head motion model is generated according to the duration of accents. Let the duration of the accent be t_a seconds. For $t_a/2$ seconds head center of mass is shifted in $+y$ direction with 25 pixels/second. For the remaining $t_a/2$ seconds head center of mass is shifted back to it's resting positions. Practical aspect of this methodology is that, the accents with short period are visually eliminated and the accents with long period are visually amplified.

VIII. CONCLUDING REMARKS AND FUTURE WORK

In this project, gesture synthesizer based an audio-visual correlation is presented. Audio-visual correlation analysis is conducted using acoustic and visual events. Acoustic events are divided into semantic and prosodic categories. Visual events are selected as hand and head gestures. The types of events are defined by investigating a portion of the database. The repetitive patterns for acoustic events are mainly keywords (*left*, *right* and *straight*) and accents. The repetitive patterns for head gestures are *nod* and *tilt*. *Left* movement of left hand,

right movement of right hand and *down* movement of both hands are defined as hand gestures.

Given a limited number of examples for each kind of event, an event model is created. Using the training portion of the database, these events are spotted and co-occurring patterns are investigated to train the correlation model. Spotting of keyword, accents and head movements was not problematic and spotting performances were high enough to be applicable. However hand movement spotting rate was not high enough since the hand movements for gestures are not well defined motions.

Investigating the co-occurring patterns, we concluded that keywords and corresponding hand movements are strongly correlated. Moreover, *nod* movement of head is found out to be highly correlated with accents. Motivated from this fact, using the test portion of the database, first, keywords and accents are detected. Then the virtual body is animated using corresponding visual event at those detected acoustic events.

As a future work, our ultimate goal is building up new audio-visual databases. The scenario of the database that is used in our project is "Direction Giving" and the scenarios can also be extended in new databases. The number of keywords and gesture patterns will be increased using new scenarios for synthesis of more natural gestures.

ACKNOWLEDGMENT

Authors would like thank Ana Huerta Carrillo and Arman Savran for their help in realizing the 3D Animation. We also thank Hannes Pirker for inspiring discussions.

REFERENCES

[1] Y. Yasinnik, M. Renwick,S. Shattuck-Hufnagel, "The Timing of Speech-Accompanying Gestures with Respect to Prosody," Conf. of The From Sound to Sense, 2004.

[2] S. Duncan, F. Parrill, D. Loehr, "Discourse factors in gesture and speech prosody," Conf. of the International Society for Gesture Studies (ISGS), Lyon, France, 2005.

[3] Jie Yao and Jeremy R. Cooperstock, "Arm Gesture Detection in a Classroom Environment," Proc. WACV'02 pp. 153-157, 2002.

[4] Y. Azoz, L. Devi. R. Sharma, "Tracking Hand Dynamics in Unconstrained Environments," Proc. Int. Conference on Automatic Face and Gesture Recognition'98 pp. 274-279, 1998.

[5] S. Malassiotis, N. Aifanti, M.G. Strintzis, "A Gesture Recognition System Using 3D Data," Proc. Int. Symposium on 3D Data Processing Visualization and Transmission'02 pp. 190-193,2002.

[6] J-M. Chung, N. Ohnishi, "Cue Circles: Image Feature for Measuring 3-D Motion of Articulated Objects Using Sequential Image Pair," Proc. Int. Conference on Automatic Face and Gesture Recognition'98 pp. 474-479, 1998.

[7] S. Kettebekov, M. Yeasin, R. Sharma, "Prosody based co-analysis for continuous recognition of coverbal gestures,"Proc. ICMI'02 pp.161-166, 2002.

[8] F. Quek, D. McNeill, R. Ansari, X-F. Ma, R. Bryll, S. Duncan, K.E. McCullough "Gesture cues for conversational interaction in monocular video," Proc. Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems'99 pp. 119-126, 1999.

[9] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound." Proc. of the Inst. of Phonetic Sciences 17: pp. 97-110, 1993.

[10] A. L. Ronzhin, Y. A. Kosarev, I.V. Lee, A. A. Karpov. "The method of continuous speech recognition based on signal analysis in a sliding window and theory of the fuzzy sets". In scientific-theoretical journal "Artificial intelligence", Donetsk, Ukraine, 2004. vol. 4. pp. 256-263.

[11] For detailed information visit: http://htk.eng.cam.ac.uk

[12] J. Caminero, C. de la Torre, L. Villarrubia, C. Martin, L. Hernindez, "On-line garbage modeling with discriminant analysis for utterance verification". In Proc. of 4-th International Conference on Spoken Language Processing ICSLP 96, Philadelphia, PA, USA, pp. 2111-2114.

[13] P. Viola and M.J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features",IEEE CVPR, 2001.

[14] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection", IEEE ICIP 2002, Vol. 1, pp. 900-903, Sep. 2002.

[15] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature TrackerDescription of the algorithm", Intel Corporation, Microprocessor Research Labs, OpenCVDocuments, 1999

[16] R. K. Mehra, "On the identification of variances and adaptive Kalman filtering", IEEE Transactions on Automatic Control, AC-15, pp. 175-183, 1970.

[17] H. Lee and J. H. Kim, "An HMM-Based Threshold Model Approach for Gesture Recognition", IEEE Transactions on Pattern Analysis Machine Intelligence, 21, 10 (Oct. 1999), 961-973.

[18] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recgonition", Proceedings of the IEEE, Vol. 77, No. 2 (February. 1989).

[19] J.Hirschberg and G.Ward, "The influence of pitch range, duration, amplitude and spectral features on the interpretation of the rise-fall-rise intonation contour in English", Journal of Phonetics, v.20, n.2, pp. 241-251, 1992.

[20] Epstein, M.A., "Voice quality and prosody in English.Proceedings of the XVth International Congress of Phonetic Sciences", ICPhS 03, 2003.

APPENDIX I
SOFTWARE NOTES

A. Codes related to head and hand gesture recognition and hand gesture modeling

For HMM training HMM routines in "Bayes Net Toolbox for Matlab, written by Kevin Patrick Murphy et al." are used. To run the codes, this BNT toolbox must be in the path of Matlab.

For head gesture related routines, check the folder head - run readHeadavi.m or readHeaddata.m or readHead2min.m to read and save data in a mat file OR write your own routine to read data. You may have to change the paths for some files - to train HMMs with continuous observations run trainHMMsCont, but first modify the name of mat file if needed - gesturespot routine runs for a sequence and spots the gestures and non gestures

For hand gesture related routines, check the folder hand - run readHanddata.m to read and save data in a mat file - trainHMMsCont trains HMM models for each gesture - traindiffHMMsCont trains different HMM models (using different length of feature vectors) for each hand gesture - plottrajectory plots the trajectories of gestures in the training set - plotartificialtrajectoryCont plots produced trajectories - generatetesttraj generates trajectories acc to the timestamps of keywords in speech